Next-Generation Storage Efficiency with Dell PowerScale Inline Data Reduction

April 2025

H17549.15

White Paper

Abstract

This paper focuses on Dell PowerScale OneFS, a modern file system that meets the unique needs of big data. OneFS includes inline data reduction, combining native, real-time data compression and deduplication. Inline data reduction enables enterprises to lower storage costs and footprint and increase data efficiency, without sacrificing data protection or management simplicity.

DELL Technologies

Copyright

Contents

Executive summary

Overview

OneFS inline data reduction combines both real-time compression and deduplication. Compression uses a lossless algorithm to reduce the physical size of data when it is written to disk and decompresses the data when it is read back. More specifically, lossless compression reduces the number of bits in each file by identifying and reducing or eliminating statistical redundancy. No information is lost in lossless compression, and a file can easily be decompressed to its original form.

Deduplication differs from data compression in that it eliminates duplicate copies of repeating data. While compression algorithms identify redundant data inside individual files and encode the redundant data more efficiently, deduplication inspects data and identifies sections, or even entire files, that are identical. Then, it replaces them with a shared copy.

Both compression and deduplication are transparent to all applications that sit on top of the file system including protocol-based services like NFS, SMB, HDFS, or S3. The primary purpose of OneFS inline data reduction is to reduce the storage requirements for data. This ability results in a smaller storage footprint, reduced power and cooling requirements, and a reduction in the overall per-TB storage cost. Also, inline data reduction helps to shrink the total amount of physical data written to storage devices. This feature is particularly beneficial for solid state drives (SSDs) and other media with finite overwrite limits, by significantly reducing flash-drive wear rates.

There are three primary measures of storage capacity that are relevant here:

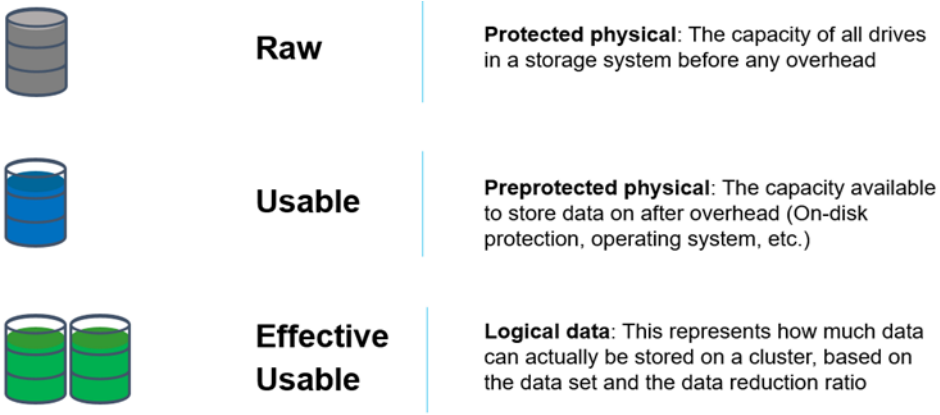

Figure 1. The primary measures of storage capacity

Savings due to inline data reduction are highly dependent on the data and can vary considerably. This variance means that accurate rates of savings are not able to be predicted without comprehensive analysis of the actual dataset. Any estimates provided in this document are for broad guidance only.

Audience

This paper presents information for deploying and managing inline data reduction on a Dell PowerScale all-flash cluster. This paper does not intend to provide a comprehensive background to the OneFS architecture.

See the OneFS Technical Overview white paper for further details on the OneFS architecture.

The target audience for this white paper is anyone configuring and managing inline data reduction in a Dell PowerScale clustered storage environment containing F910, F900, F810, F710, F600, F210, F200, H700/7000, H5600, or A300/3000 nodes. It is assumed that the reader has an understanding and working knowledge of the OneFS components, architecture, commands, and features.

More information about OneFS commands and feature configuration is available in the OneFS Administration Guide.

Glossary of terms

Table 1. Glossary of terms

Term	Description
Compression	The process of modifying, encoding, or converting the bits structure of data in such a way that it consumes less space on the storage media.
Deduplication	The elimination of duplicate or redundant data, thereby lowering the actual physical storage required.
Inflation	Uncompression of data.
Rehydration	Un-deduplication of data.
Protection Group (PG)	OneFS embeds protection into each individual file. To do this, files are segmented into sections, or protection groups (PGs), which are independently protected. The PG tracks all the information about how that logical piece of the file is physically stored, and protected, on-disk.
Cluster	In protection group terminology, a cluster is a division of the logical-block space with IFS MAXCONTIG size and alignment.
Compression Chunk Size	The maximum amount of logical file data that is compressed as a single entity (128 KB for OneFS). When an I/O occurs to a compressed region, this also represents the smallest possible I/O, since the I/O must be expanded to include the entire compression chunk.
Packing	The combining of multiple compression chunks together to reduce lost space.
Write Amplification	The cost of additional file system writes to the storage device due to the compression implementation.
Hardware offload	Compression is efficiently performed by a dedicated FPGA on a PCIe card, rather than using the node's CPU cycles.
Inline Compression	Data is compressed and decompressed on the fly as it is written to and read from to disk
Post-process Compression	Data is compressed in a second pass after it has already been written to disk.
Lossless compression	Reduction of a file's size with no discernable loss of quality. Lossless compression rewrites the data of the original file in a more efficient way.
Inline Deduplication	Data is deduplicated on the fly as it is written to disk.

Term	Description
Post-process Deduplication	Data is deduplicated in a second pass after it has already been written to disk.
Zero Block Removal	Blocks that contain only zeros are detected and prevented from being written to disk.
Raw	Equivalent to 'Protected physical'. Total footprint of data including protection overhead FEC erasure coding) and excluding data efficiency savings (compression and deduplication).
Usable	Equivalent to 'Preprotected Physical'. Data size excluding protection overhead and including data efficiency savings (compression and deduplication).
Effective	Equivalent to 'Logical data'. Data size excluding protection overhead and excluding data efficiency savings (compression and deduplication).
Logical Data	Equivalent to 'Effective'. Data size excluding protection overhead and excluding data efficiency savings (compression and deduplication).
Dedupe Saved	Capacity savings from deduplication.
Compression Saved	Capacity savings from inline compression.
Preprotected Physical	Equivalent to 'Usable'. Data size excluding protection overhead and including data efficiency savings (compression and deduplication).
Protection Overhead	Size of erasure coding used to protect data.
Protected Physical	Equivalent to 'Raw'. Total footprint of data including protection overhead FEC erasure coding) and excluding data efficiency savings (compression and deduplication).
Effective to Raw Ratio	Compression ratios stated as Effective to Raw are calculated including the data's protection overhead. OneFS compression ratios are typically calculated and reported using this method.
Effective to Usable Ratio	Compression ratios stated as Effective to Usable are calculated omitting protection overhead. Competitors' compression ratios are often calculated and reported using this method.
COW	Copy on write, to preserve data in snapshots.
EC	Endurant cache.
BAM	Block allocation manager.
BSW	BAM safe write.
Coalescer	OneFS non-volatile write cache.

Revisions

Date	Part number/ revision	Description
February 2019	H17549	Updated for OneFS 8.1.3
April 2019	H17549.1	Updated for OneFS 8.2.0
August 2019	H17549.2	Updated for OneFS 8.2.1

Date	Part number/ revision	Description
December 2019	H17549.3	Updated for OneFS 8.2.2
May 2020	H17549.4	Updated for OneFS 9.0
September 2020	H17549.5	Updated for OneFS 9.1
April 2021	H17549.6	Updated for OneFS 9.2
September 2021	H17549.7	Updated for OneFS 9.3
March 2022	H17549.8	Updated for OneFS 9.4
January 2023	H17549.9	Updated for OneFS 9.5
February 2024	H17549.10	Updated for OneFS 9.7
April 2024	H17549.11	Updated for OneFS 9.8
May 2024	H17549.12	Updated for PowerScale F910
August 2024	H17549.13	Updated for OneFS 9.9
December 2024	H17549.14	Updated for OneFS 9.10
April 2025	H17549.15	Updated for OneFS 9.11

We value your feedback

Dell Technologies and the authors of this document welcome your feedback on this document. Contact the Dell Technologies team by email.

Author: Nick Trimbee

Note: For links to other documentation for this topic, see the PowerScale Info Hub.

Inline data reduction architecture

Overview

OneFS data reduction offers both inline data compression and inline deduplication, and the supporting OneFS architecture is comprised of the following principal components:

- Data Reduction Platform
- Compression Engine and Chunk Map
- Zero block removal phase
- Deduplication In-memory Index and Shadow Store Infrastructure
- Data Reduction Alerting and Reporting Framework
- Data Reduction Control Path

The inline data reduction control path consists of the OneFS command-line interface (CLI) and RESTful platform API and is responsible for managing the configuration and reporting of the feature.

Data reduction platform

Inline data reduction is supported on the Dell PowerScale F910, F900, F710, and F600, and F210-NVMe and F200 SSD nodes, PowerScale H700/7000 and A300/3000 node, and the Isilon F810 and H5600 platforms.

The specific OneFS versions required to support a cluster or node pool with the following characteristics include:

- OneFS 8.2.1 or later for an F810 node pool.
- OneFS 8.2.2 or later for an H5600 node pool.
- OneFS 9.0 or later for an F600 or F200 node pool.
- OneFS 9.2 or later for an F900 node pool.
- OneFS 9.2.1 or later for an H700, H7000, A300, or A3000 node pool.
- OneFS 9.7 or later for an F710 of F210 node pool.
- OneFS 9.8 or later for an F910 node pool.

Unlike the other platforms above, each F810 node includes a data reduction hardware offload adapter. This adapter off-loads certain tasks from the CPU. Specifically, data compression and inflation are transparently performed by the offload adapter with minimal latency, avoiding the need for consuming a node's expensive CPU and memory resources.

Each F810 node's data reduction offload adapter contains an FPGA chip, which is dedicated to the compression of data received over client connections to the node. These cards reside in the backend PCIe slot in each of the four nodes. The two Ethernet ports in each adapter are used for the node's redundant backend network connectivity.

Data reduction workflow

Data from network clients is accepted as is and makes its way through the OneFS write path until it reaches the BSW engine, where it is broken up into individual chunks. The inline data reduction write path consists of three main phases:

- Zero Block Removal
- Inline Deduplication
- Inline Compression

If both inline compression and deduplication are enabled on a cluster, zero block removal is performed first, followed by deduplication, and then compression. This order allows each phase to reduce the scope of work each subsequent phase.

Figure 2. Inline data reduction workflow

Zero block removal

The inline data reduction zero block removal phase detects blocks that contain only zeros and prevents them from being written to disk. This action both reduces disk space requirements and avoids unnecessary writes to SSD, resulting in increased drive longevity.

Zero block removal occurs first in the OneFS inline data reduction process. As such, it has the potential to reduce the amount of work that both inline deduplication and compression need to perform. The check for zero data does incur some overhead. However, for blocks that contain non-zero data the check is terminated on the first non-zero data found, which helps to minimize the impact.

The following characteristics are required for zero block removal to occur:

- A full 8 KB block of zeroes
- A partial block of zeroes being written to a sparse or preallocated block

The write will convert the block to sparse if not already. A partial block of zeroes being written to a non-sparse, non-preallocated block will not be zero eliminated.

Inline deduplication

With inline data reduction, deduplication is performed in real time, as data is written to the cluster. Storage efficiency is achieved by scanning the data for identical blocks as it is received and then eliminating the duplicates.

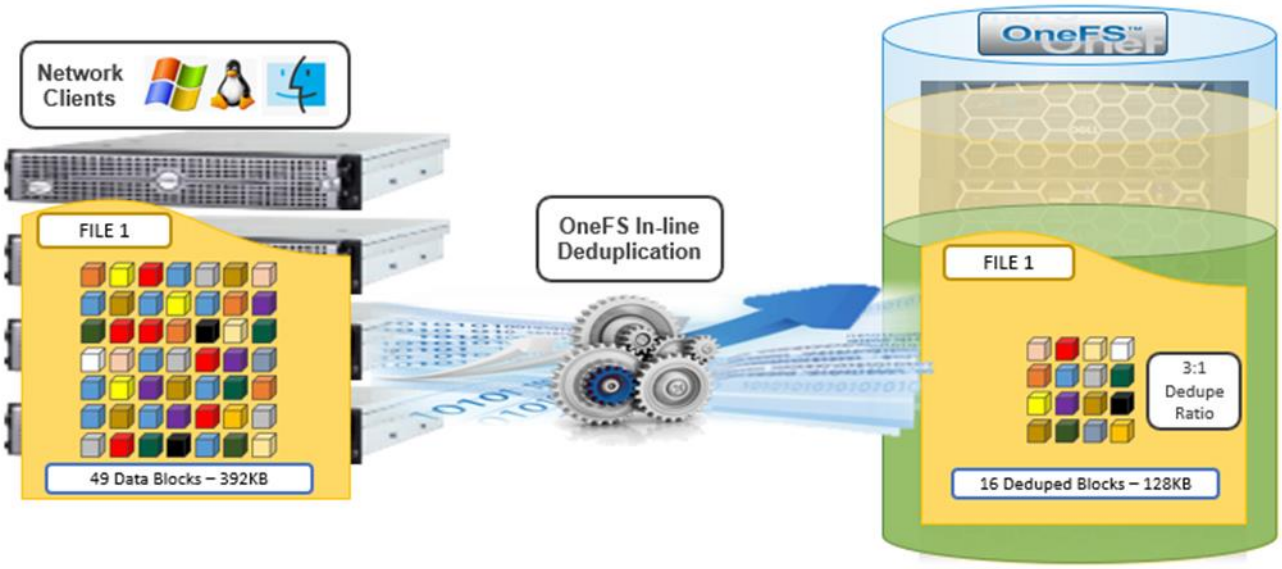

Figure 3. OneFS inline deduplication

When a duplicate block is discovered, inline deduplication moves a single copy of the block to a special set of files known as shadow stores. OneFS shadow stores are file system containers that allow data to be stored in a shareable manner. As such, files on OneFS can contain both physical data and pointers, or references, to shared blocks in shadow stores.

Shadow stores were initially introduced to support OneFS file clones, and there are many overlaps between cloning and deduplicating files. The other main consumer of shadow stores is OneFS Small File Storage Efficiency. This feature maximizes the space utilization of a cluster by decreasing the amount of physical storage required to house the small files that consist of a typical healthcare dataset.

Shadow stores are similar to regular files but are hidden from the file system namespace, so they cannot be accessed by a pathname. A shadow store typically grows to a maximum size of 2 GB, which is around 256 K blocks, with each block able to be referenced by 32,000 files. If the reference count limit is reached, a new block is allocated, which may or may not be in the same shadow store. Also, shadow stores do not reference other shadow stores. And snapshots of shadow stores are not permitted because the data contained in shadow stores cannot be overwritten.

When a client writes a file to a node pool configured for inline deduplication on a cluster, the write operation is divided up into whole 8 KB blocks. Each of these blocks is hashed, and its cryptographic ‘fingerprint’ is compared against an in-memory index for a match. At this point, one of the following operations will occur:

1. If a match is discovered with an existing shadow store block, a byte-by-byte comparison is performed. If the comparison is successful, the data is removed from the current write operation and replaced with a shadow reference.
2. When a match is found with another LIN, the data is written to a shadow store instead and replaced with a shadow reference. Next, a work request is generated and queued that includes the location for the new shadow store block, the matching

LIN and block, and the data hash. A byte-by-byte data comparison is performed to verify the match, and the request is processed.

3. If no match is found, the data is written to the file natively and the hash for the block is added to the in-memory index.

In order for inline deduplication to be performed on a write operation, the following conditions need to be true:

- Inline deduplication must be globally enabled on the cluster.
- The current operation is writing data (not a truncate or write zero operation).
- The 'no_dedupe' flag is not set on the file.
- The file is not a special file type, such as an alternate data stream (ADS) or an EC (endurant cache) file.
- Write data includes fully overwritten and aligned blocks.
- The write is not part of a 'rehydrate' operation.
- The file has not been packed (containerized) by SFSE (small file storage efficiency).

OneFS inline deduplication uses the 128-bit CityHash algorithm, which is both fast and cryptographically strong. This contrasts with the OneFS post-process SmartDedupe, which uses SHA-1 hashing.

Each F910, F900, F810, F600, F200, F700/7000, H5600, or A300/3000 node in a cluster with inline deduplication enabled has its own in-memory hash index that it compares block 'fingerprints' against. The index is in system RAM, is allocated using physically contiguous pages, and is accessed directly with physical addresses. This system avoids the need to traverse virtual memory mappings and does not incur the cost of translation lookaside buffer (TLB) misses, minimizing deduplication performance impact.

The maximum size of the hash index is governed by a pair of sysctl settings, one of which caps the size at 16 GB, and the other which limits the maximum size to 10% of total RAM. The strictest of these two constraints applies. While these settings are configurable, the recommended best practice is to use the default configuration. Any changes to these settings should only be performed under the supervision of Dell support.

Since inline deduplication and SmartDedupe use different hashing algorithms, the indexes for each are not shared directly. However, the work performed by each deduplication solution can be used by each other. For instance, if SmartDedupe writes data to a shadow store, when those blocks are read, the read hashing component of inline deduplication will see those blocks and index them.

When a match is found, inline deduplication performs a byte-by-byte comparison of each block to be shared to avoid the potential for a hash collision. Data is prefetched before the byte-by-byte check and then compared against the L1 cache buffer directly, avoiding unnecessary data copies and adding minimal overhead. Once the matching blocks have been compared and verified as identical, they are shared by writing the matching data to a common shadow store and creating references from the original files to this shadow store.

Figure 4. OneFS duplicate block sharing

Inline deduplication samples every whole block written and handles each block independently, so it can aggressively locate block duplicity. If a contiguous run of matching blocks is detected, inline deduplication will merge the results into regions and process them efficiently.

Inline deduplication also detects deduplication opportunities from the read path, and blocks are hashed as they are read into L1 cache and inserted into the index. If an existing entry exists for that hash, inline deduplication detects there is a block-sharing opportunity between the block that it most recently read and the one previously indexed. It combines that information and queues a request to an asynchronous deduplication worker thread. As such, it is possible to deduplicate a dataset purely by reading it all. To help mitigate the performance impact, the hashing is performed out-of-band in the prefetch path, rather than in the latency-sensitive read path.

Inline compression

The F810 nodes use an FPGA-based hardware offload engine resident on the back-end PCIe network adapter to perform real-time data compression. This action occurs as files are written to from a node in the cluster using a connected client session. Similarly, files are re-inflated on demand as they are read by clients.

On top of the FPGA, the OneFS hardware offload engine uses a proprietary implementation of DEFLATE with the highest level of compression, while incurring minimal to no performance penalty for highly compressible datasets.

The compression engine consists of three main components:

Table 2. OneFS data reduction engine components

Engine component	Description
Search Module	LZ77 search module analyzes inline file data chunks for repeated patterns.
Encoding Module	Performs data compression (Huffman encoding) on target chunks.
Decompression Module	Regenerates the original file from the compressed chunks.

Since they reside on the same card, the data compression engine shares PCIe bandwidth with the node's backend Ethernet interfaces. In general, there is plenty of bandwidth

available. A best practice is to run highly compressible datasets through the F810 nodes with compression enabled. However, it is not advisable to run non-compressible datasets with compression enabled.

OneFS provides software-based compression for the F910, F900, F710, F600, F210, F200, H700/7000, H5600, and A300/3000 platforms. Compression in software is also used as fallback in the event of an F810 hardware failure, and in a mixed cluster for use in nodes without a hardware offload capability. Both hardware and software compression implementations are DEFLATE compatible.

Compression file chunking

When a file is written to OneFS using inline data compression, the file's logical space is divided up into equal-sized chunks called compression chunks. Compaction is used to create 128 KB compression chunks, with each chunk consisting of sixteen 8 KB data blocks. This setup is optimal since 128 KB is the same chunk size that OneFS uses for its data protection stripe units, providing simplicity and efficiency, and avoids the overhead of additional chunk packing.

For example, consider the following 128 KB chunk:

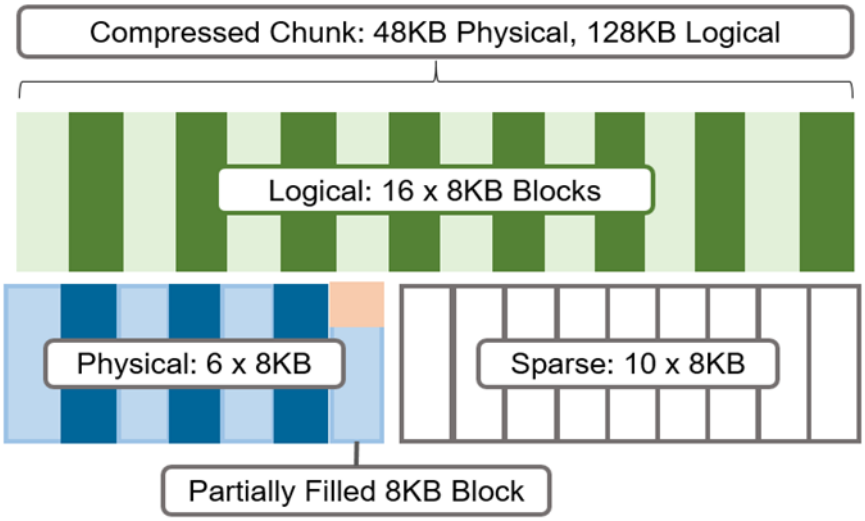

Figure 5. Compression chunks and OneFS transparent overlay

After compression, this chunk is reduced from sixteen to six 8 KB blocks in size. This reduction means that this chunk is now physically 48 KB in size. OneFS provides a transparent logical overlay to the physical attributes. This overlay describes whether the backing data is compressed or not and which blocks in the chunk are physical or sparse, such that file system consumers are unaffected by compression. As such, the compressed chunk is logically represented as 128 KB in size, regardless of its actual physical size. The orange sector in the illustration above represents the trailing, partially filled 8 KB block in the chunk. Depending on how each 128KB chunk compresses, the last block may be under-utilized by up to 7KB after compression.

Efficiency savings must be at least 8 KB (one block) for compression to occur, otherwise that chunk or file will be passed over and remain in its original, uncompressed state. For example, a file of 16 KB that yields 8 KB (one block) of savings would be compressed. Once a file has been compressed, it is protected with Forward Error Correction (FEC)

parity blocks, reducing the number of FEC blocks and therefore providing further overall storage savings.

Compression chunks will never cross node pools, avoiding the requirement to decompress or recompress data to change protection levels, perform recovered writes, or otherwise shift protection-group boundaries.

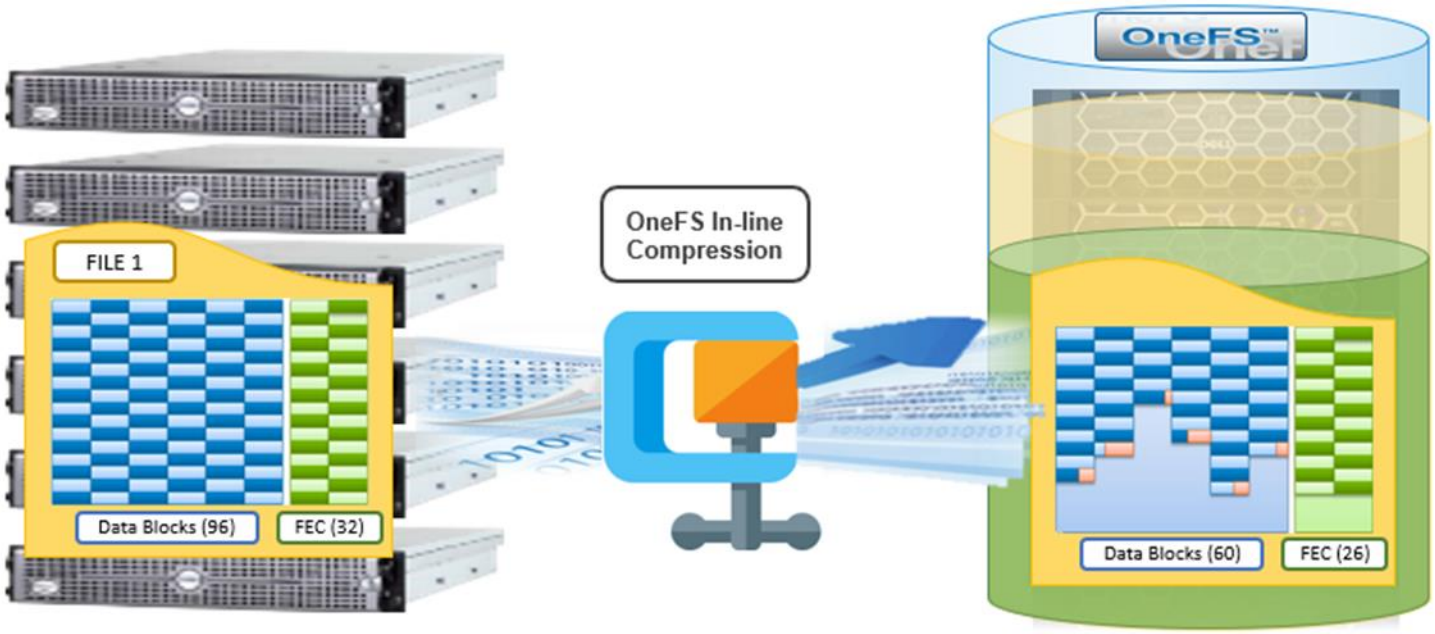

Figure 6. OneFS inline compression

In the illustration above, a 768 KB file (file 1) is written from a Windows client to an F810 cluster. After passing through the OneFS inline compression engine, the logical data footprint of that file is reduced from ninety-six to sixty 8 KB blocks, across six chunks. This is represented by the blue data blocks. Then, the file is FEC protected using twenty-six parity blocks, shown in green.

Data reduction write path

In a PowerScale cluster, data, metadata, and inodes are all distributed across multiple drives on multiple nodes. When reading or writing to the cluster, a client is directed by SmartConnect to the desired protocol head on a particular node, or initiator. This node then acts as the 'captain' for the operation. It performs the chunking and data compression, orchestrates the layout of data and metadata, creates erasure codes, and performs normal operations of lock management and permissions control.

For example, assume you have a four-node F810 cluster. A Windows client connects to the top half, or initiator, on node 1 to write a file. After the SMB session is established and write request granted, the client begins to send the file data across the front-end network to the cluster where it is initially buffered in the coalescer, or OneFS write cache. The purpose of the coalescer is to build up a large contiguous range of data that will make the write operation more efficient.

When the coalescer is flushed, data chunks, typically sized on protection group boundaries, are passed through the data reduction pipeline. First, if inline deduplication is enabled, the incoming data is scanned for zero block removal and deduplication

opportunities. When found, any zero blocks are stripped out and any matching blocks are deduplicated. Next, chunks that meet the 'compressibility' criteria described above are compressed by the FPGA. Finally, the initiator runs its 'write plan' for the file data. This action optimizes for layout efficiency and the selected protection policy, and the chunks/stripes are written to SSDs on the bottom half of the participant nodes.

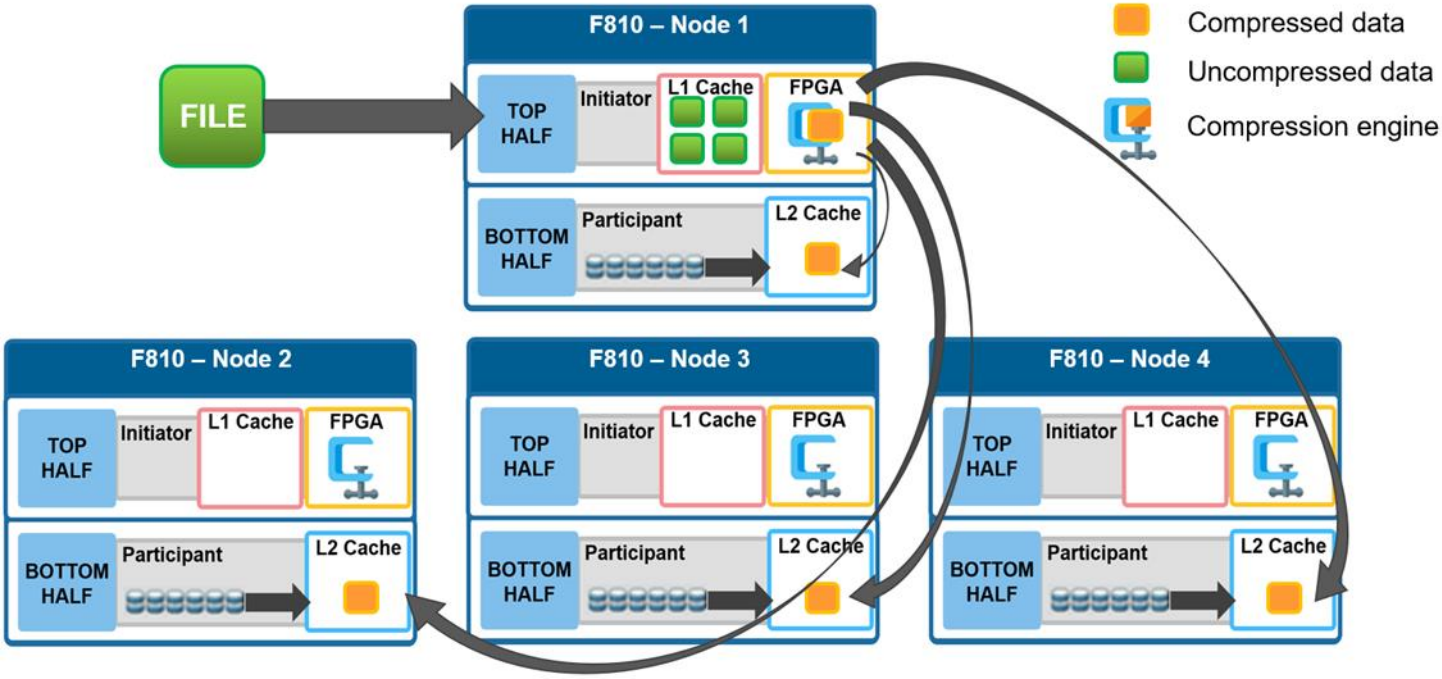

Figure 7. File writes with compression

OneFS stripes data across all nodes—and not simply across disks—and protects the files, directories, and associated metadata using software erasure-code or mirroring technology. Erasure coding can provide beyond 80% efficiency on raw disk with five nodes or more, and on large clusters can even do so while providing quadruple-level redundancy. For any given file, the stripe width is the number of nodes (not disks) that a file is written across. For example, on the 4-node F810 cluster above with the recommended +2d:1n protection level, OneFS will use a stripe width of 8 and protection level of 6+2, where each node is used twice: Two data stripe units are written to each of three nodes, and two FEC units to the remaining node.

For further details about OneFS data protection, see the OneFS Technical Overview white paper.

Data reduction read path and caching integration

In the diagram below, an NFS client attaches to node 1 and issues a read request for a file. Node 1, the captain, gathers all the chunks of data from the various nodes in the cluster and presents it in a cohesive way to the requesting client. Since the file's data has been stored in a compressed form on nodes' SSDs, node 1 needs to gather all the constituent chunks and decompress the data so the file can be sent across the wire to the client in its original form.

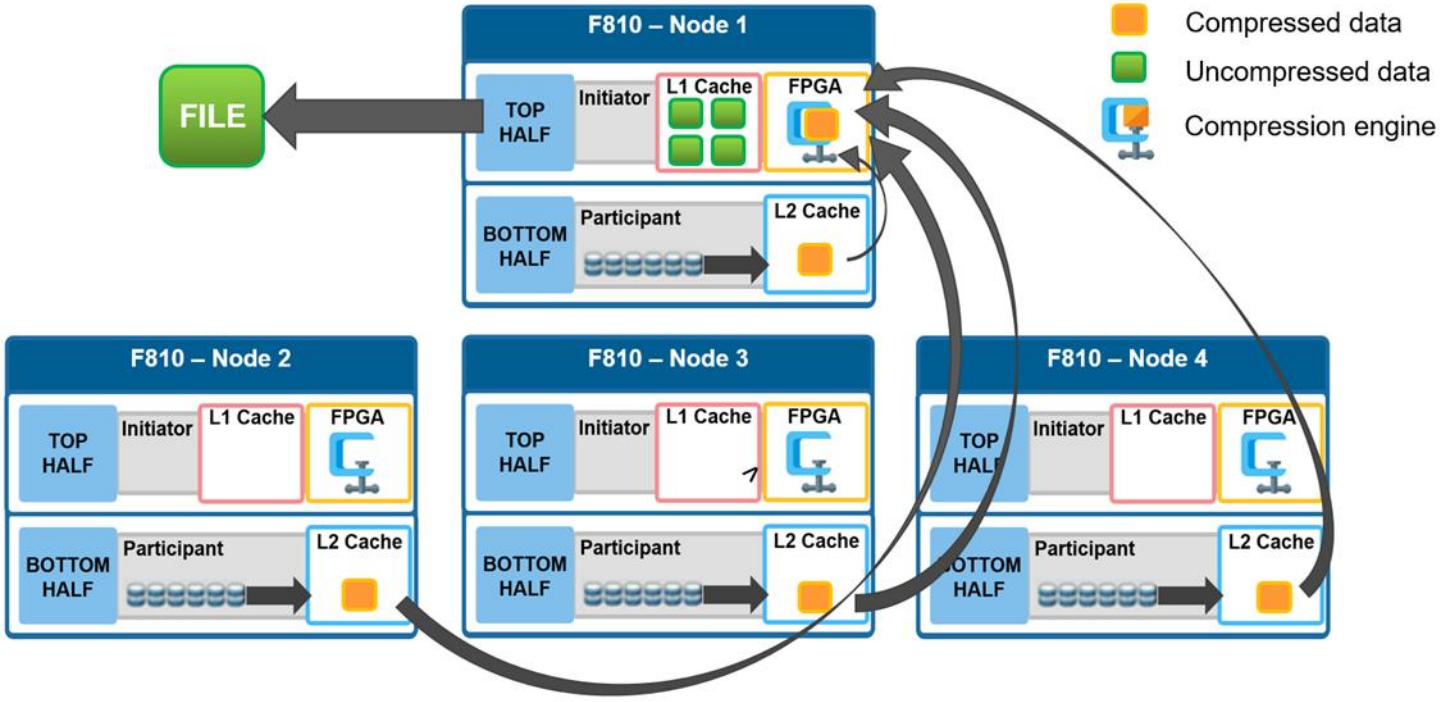

Figure 8. File reads with compression

During this read process, the L2 read cache on the participant nodes (nodes 2-4) is populated with the compressed data chunks that are sent to node 1. This means that any additional read requests for this file can be served straight from low latency cache, rather than reading again from the drives. This process both accelerates read performance and reduces wear on the SSDs.

To support OneFS inline compression, a node's L1, or client-side, read cache is divided into separate address spaces so that both the on-disk compressed data and the logical uncompressed data can be cached. The address space for the L1 cache is already split for data and FEC blocks, so a similar technique is used to divide it again. Data in the uncompressed L1 cache is fed from data in the compressed L1 cache which, in turn, is fed from disk.

OneFS prefetch caching has also been enhanced to accommodate compressed data. Since reading part of a compressed chunk results in the entire compression chunk being cached, it will effectively mean that prefetch requests are rounded to compression chunk boundaries. Since a prefetch request is not complete until the uncompressed data is available in cache, the callback used for prefetch requests performs the decompression.

Data reduction in a mixed cluster

A mixed, or heterogeneous, cluster is one that comprises two or more different types or node. For compression, there are two main concepts in play in a mixed cluster:

4. Reading compressed data to return the logical data (for client traffic, NDMP, or otherwise).
5. Writing a previously compressed file to disk in uncompressed format (because the target tier does not support compression).

The former happens in L1 memory and not on-disk. As such, only F910, F900, F810, F600, F200, F700/7000, H5600, and A300/3000 storage pools may contain compressed data.

In general, OneFS does not allow deduplication across different disk pool policies. For inline deduplication, a deduplicated file that is moved to another tier will retain the shadow references to the shadow store on the original disk pool. While this behavior violates the rule for deduping across different disk pool policies, it is preferred to do this than rehydrate files that are moved. Further deduplication activity on that file will no longer be allowed to reference any blocks in the original shadow store. The file will need to be deduplicated against other files in its new node pool.

Writes to a mixed cluster

Figure 9 below depicts a file write in a mixed cluster environment. A client connects to an H400 node and writes a file with a path-based file pool policy that directs the file to an F810 nodepool. In this scenario, the H400 node first scans the incoming data for zero block removal or deduplication opportunities.

Zero block elimination and inline deduplication will only be enabled on nodes that have local drives in a disk pool with the data reduce flag set.

The data reduce flag will only be set on F910, F900, F810, F710, F600, F210, F200, H700/7000, H5600, and A300/3000 disk pools and therefore zero block elimination and inline deduplication will only be performed on those platforms. When found, any zero blocks are stripped out and any matching blocks are deduplicated.

Next, the data is divided into 128 KB compression chunks as usual. However, since the H400 node does not have an FPGA offload card, it instead performs compression on the chunks in software.

Unlike inline deduplication, compression does not require the initiator node to be a member of a disk pool with the data reduce flag set. If the target disk pool for the write has data reduction set and the cluster has inline compression enabled, compression will be performed.

A different compression algorithm is used to help minimize the performance impact. Each compressed chunk is then FEC protected and the H400 uses its write plan to place blocks on the participant nodes. The chunks are written in compressed form over the back-end network to the appropriate F810 nodes.

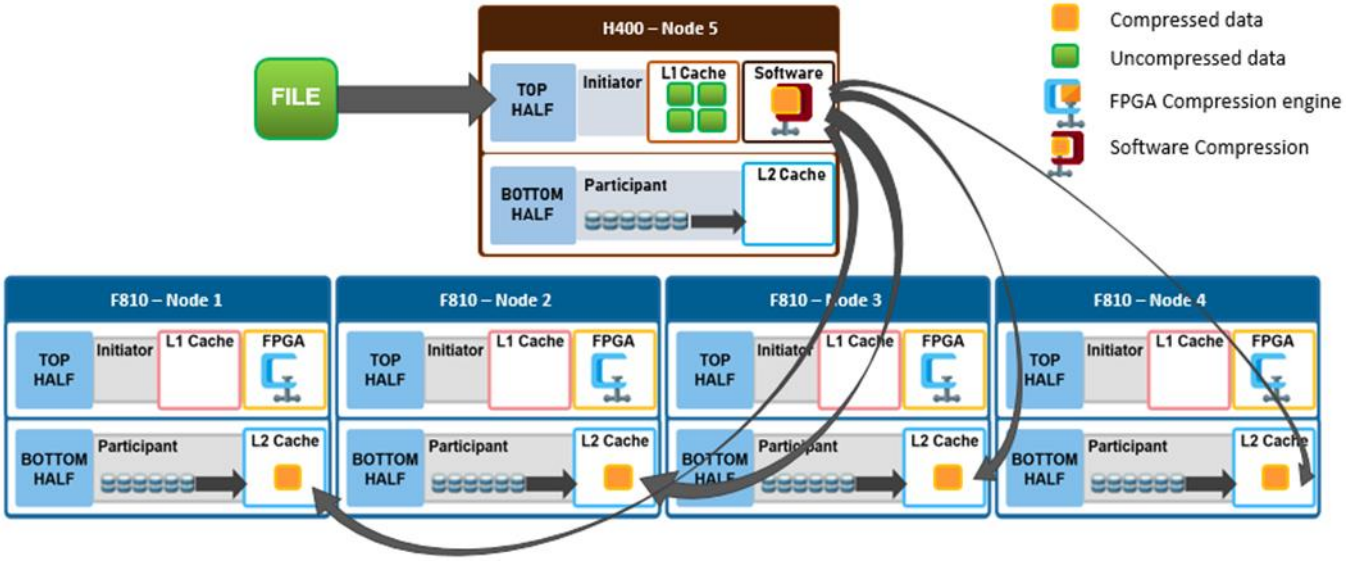

Figure 9. File write in a mixed cluster with software compression

Reading from a mixed cluster

In the following mixed cluster scenario, a client connects to an H400 node and issues a read request for a compressed file housed on an F810 node pool. The H400 retrieves all the compressed chunks from the pertinent F810 nodes over the backend network. Since the H400 has no FPGA offload card, the decompression of the chunks is performed in software. Software compression uses a different DEFLATE-compatible algorithm to help minimize the performance impact of non-offloaded decompression. After the chunks have been decompressed, the file is reassembled and sent over Ethernet in uncompressed form to the requesting client.

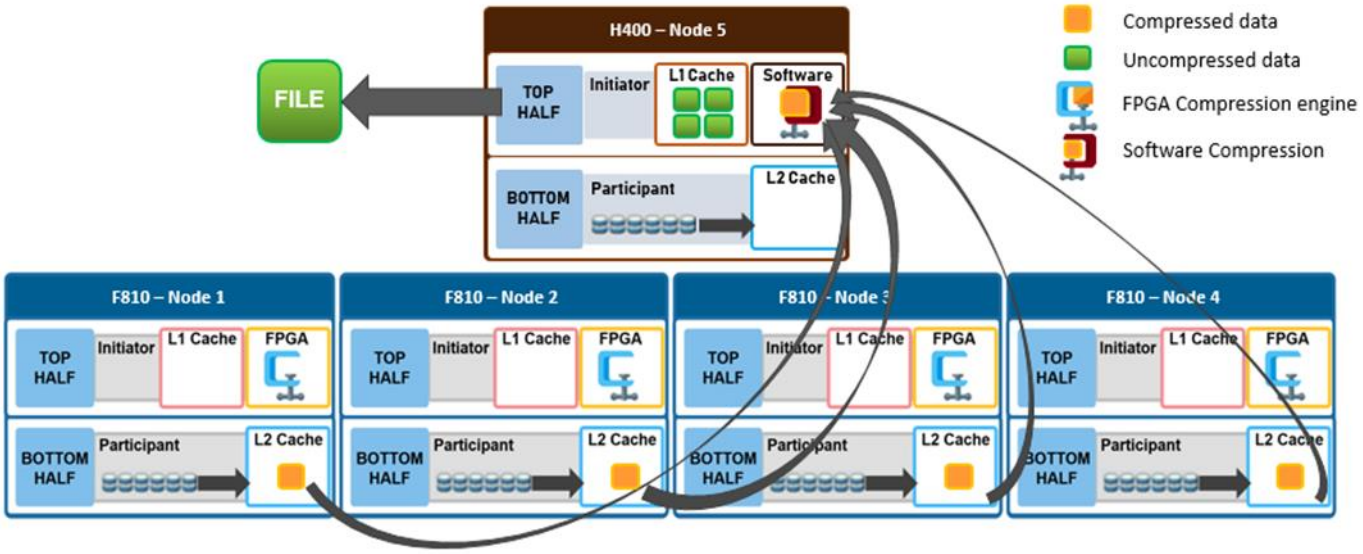

Figure 10. File read in a mixed cluster with software compression

Data reduction and tiering in a mixed cluster

Consider a mixed cluster consisting of an F810 flash performance tier and an A2000 archive tier. SmartPools is licensed, and a file pool policy is crafted to move data over three months old from the F810 flash tier to the archive tier. Files are stored in a compressed form on the F810 tier.

When SmartPools runs, decompression occurs as the files targeted for down-tiering are restriped from the F810 tier to the A2000 tier inside the SmartPools job. The SmartPools job runs across all the nodes in the cluster, both the F810 and A2000, and the availability of hardware assisted decompression depends on which node or nodes are running the job worker threads. After files are on the A2000 tier, they remain uncompressed.

If a file has been deduplicated, whether by inline or post process deduplication, the deduplicated state of the file will remain unchanged when tiered. This is true even if the file is moved from a disk pool that supports data reduce to a disk pool that does not.

If another file pool policy is created to up-tier files from the A2000 back up to the F810, the file chunks will be compressed when they are restriped onto the F810 nodes. Once again, all nodes in the cluster will participate in the SmartPools job.

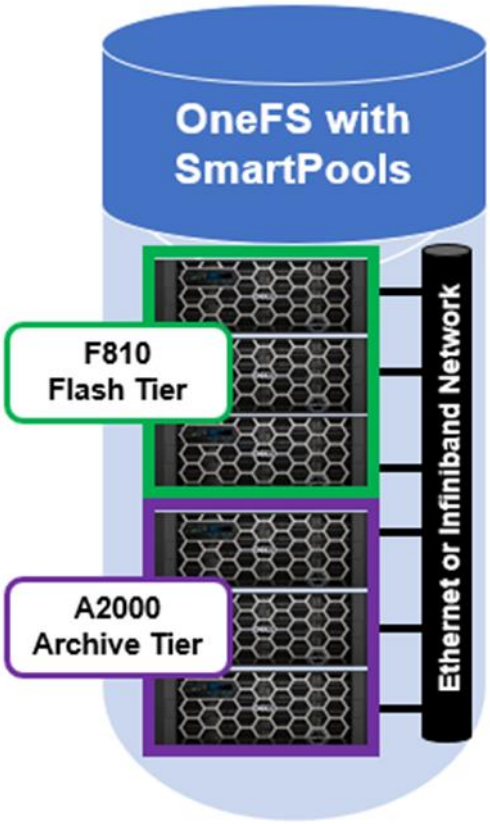

Figure 11. A SmartPools cluster with an all-flash performance tier and an archive tier

As data moves between tiers in a mixed cluster, it must be re-written from the old to new drives. If the target tier has a different ‘data-reduce’ setting than the source tier, data is compressed or decompressed as appropriate. The sizing of the F810 pool should be independent of whether it is participating in a mixed cluster or not. However, because up-tiering using the job-engine can run job tasks on the lower tier nodes, the achievable compression ratios may be slightly less efficient when software compression is used.

Data reduction and replication in a mixed cluster

SyncIQ is licensed on a mixed F810 and A2000 cluster and a SyncIQ policy is configured to replicate data to a target cluster. On the source cluster, replication traffic is isolated to only the A2000 nodes. When SyncIQ is run, worker threads on the A2000 nodes gather all the compressed chunks from the F810 nodes over the backend network (RBM). Then, the A2000 nodes perform decompression of the chunks in software. As discussed previously, software compression uses a different DEFLATE-compatible algorithm from hardware offload to help minimize the performance impact of non-offloaded decompression. After the chunks have been decompressed, the data is sent over Ethernet to the target cluster in uncompressed form.

Similarly, deduplicated data is always rehydrated when it exits a cluster. For a data service such as SyncIQ, data is replicated in its entirety and shadow stores and shadow links are not preserved on the target. This means that the target cluster must have sufficient space to house the full size of the replicated dataset. If the target cluster happens to also be F810 hardware and inline data reduction is enabled, compression or deduplication (or both) will be performed as the replication data is ingested by each target node.

Compression and backup in a mixed cluster

A mixed F810 and A2000 cluster is configured for NDMP backup from the A2000 nodes. When a backup job runs, the A2000 nodes retrieve all the compressed chunks from the pertinent F810 nodes over the backend network (RBM). Since the A2000 has no FPGA offload card, the decompression of the chunks is performed in software. Once the chunks have been decompressed, each file is reassembled and sent over Fibre Channel (2-way NDMP) or Ethernet (3-way NDMP) in uncompressed form to the backup device or devices.

With NDMP, deduplicated data is rehydrated when it leaves the cluster, and shadow stores and shadow links are not preserved on the backup. The NDMP tape device or VTL will need to have sufficient space to house the full size of the dataset.

Inline data reduction configuration and management

Overview

OneFS inline data reduction uses a simple administrative control plane. Configuration is through the command-line interface (CLI), using the ‘isi compression’ and ‘isi dedupe inline’ commands. There are also utilities provided to decompress, or rehydrate, compressed and deduplicated files if necessary. Plus, there are tools for viewing on-disk capacity savings that inline data reduction has generated.

The ‘isi_hw_status’ CLI command can be used to confirm and verify the node or nodes in a cluster. For example:

```
# isi_hw_status –i | grep Product
Product: F810-4U-Single-256GB-1x1GE-2x40GE SFP+-24TB SSD
```

Enabling compression

Since Compression configuration is binary, either on or off across a cluster, it can be easily controlled using the OneFS command-line interface (CLI). For example, the following syntax will enable compression and verify the configuration:

```
# isi compression settings view
    Enabled: No
# isi compression settings modify --enabled=True
# isi compression settings view
    Enabled: Yes
```

Note: Inline compression is enabled by default on the following:
-- New F810 clusters running OneFS 8.2.1 and later
-- New H5600 clusters running OneFS 8.2.2 and later
-- New F600 and F200 clusters running OneFS 9.0 and later
-- F900 clusters running OneFS 9.2 and later
-- H700/7000 and A300/3000 clusters running OneFS 9.2.1 and later
-- F710 and F210 clusters running OneFS 9.7 and later
-- F910 clusters running OneFS 9.8 and later

In a mixed cluster containing other node styles in addition to compression nodes, files will only be stored in a compressed form on F910, F900, F810, F710, F600, F210, F200, H5600, H700/7000, and A300/3000 node pool or pools. Data that is written or tiered to storage pools of other hardware styles will be uncompressed at that time when it moves between pools. A node on the cluster that does not support inline compression can be an initiator for compressed writes in software to a compression node pool. However, this configuration may generate significant CPU overhead for lower powered nodes, such as the A-series hardware and provide only software fallback based compression with lower compressibility.

Verifying compression

While there are no visible userspace changes when files are compressed, the ‘isi get’ CLI command provides straightforward method to verify whether a file is compressed. If compression has occurred, both the ‘disk usage’ and the ‘physical blocks’ metric reported by the ‘isi get –DD’ CLI command will be reduced. Also, at the bottom of the command’s output, the logical block statistics will report the number of compressed blocks. For example:

```
Metatree logical blocks:
   zero=260814 shadow=0 ditto=0 prealloc=0 block=2 compressed=1328
```

For more detailed information, the –O flag, which displays the logical overlay, can be used with the ‘isi get’ command. This command is described in more detail later in this paper.

Disabling compression

OneFS inline data compression can be disabled from the CLI with the following syntax:

```
# isi compression settings modify --enabled=False
# isi compression settings view
    Enabled: No
```

Enabling inline deduplication

Since inline deduplication configuration is binary, either on or off across a cluster, it can be easily controlled using the OneFS command-line interface (CLI). For example, the following syntax will enable inline deduplication and verify the configuration:

```
# isi dedupe inline settings view
    Mode: disabled
```

```
# isi dedupe inline settings modify --mode enabled
# isi dedupe inline settings view
    Mode: enabled
```

Note: Inline deduplication is enabled by default for new clusters running OneFS 9.4 and supporting inline data reduction. For earlier OneFS releases, inline deduplication is disabled by default. When upgrading to OneFS 9.4, a cluster's existing inline deduplication configuration is preserved.

Table 3. OneFS inline compression and deduplication defaults

Cluster configuration	Inline deduplication	Inline compression
New cluster running OneFS 9.4	Enabled	Enabled
New cluster running OneFS 9.3 or earlier	Disabled	Enabled
Cluster with inline deduplication enabled that is upgraded to OneFS 9.4	Enabled	Enabled
Cluster with inline deduplication disabled that is upgraded to OneFS 9.4	Disabled	Enabled

Verifying inline deduplication

While there are no visible userspace changes when files are deduplicated, if deduplication has occurred, both the 'disk usage' and the 'physical blocks' metric reported by the 'isi get –DD' CLI command will be reduced. Also, at the bottom of the command's output, the logical block statistics will report the number of shadow blocks. For example:

```
Metatree logical blocks:
   zero=260814 shadow=362 ditto=0 prealloc=0 block=2 compressed=0
```

Disabling inline deduplication

OneFS inline data deduplication can be disabled from the CLI with the following syntax:

```
# isi dedupe inline settings modify --mode disabled
# isi dedupe inline settings view
    Mode: disabled
```

Pausing inline deduplication

OneFS inline data deduplication can be paused from the CLI with the following syntax:

```
# isi dedupe inline settings modify --mode paused
```

Assess mode

OneFS inline data deduplication can be run in assess mode from the CLI with the following syntax:

```
# isi dedupe inline settings modify --mode assess
```

Events and alerts

Issues with inline compression may generate the following OneFS events and alerts. These include:

Table 4. OneFS inline compression events and alerts

Event category	Alert condition	Event trigger	Event ID
Health	Inline compression has failed on the specific node	Falling back to software	40070001
Health	Inline compression hardware is unhealthy	Increased error rates Device is delisted	900160101
Availability	Inline compression hardware is unavailable	The device is missing	900160100

Similarly, problems with inline deduplication may generate the following OneFS events and alerts. These include:

Table 5. OneFS inline deduplication events and alerts

Event category	Alert condition	Event ID
Health	Inline deduplication index allocation failed	400180001
Health	Inline deduplication index allocation in progress	400180002
Availability	Inline deduplication not supported	400180003
Health	Inline deduplication index is smaller than requested	400180004
Health	Inline deduplication index has non-standard layout	400180005

In the event that inline deduplication encounters an unrecoverable error, it will restart the write operation with inline deduplication disabled. If any of the above alert conditions occur, contact Dell Technical Support for further evaluation.

Inline data reduction efficiency

Overview

Compression and deduplication can significantly increase the storage efficiency of data. However, the actual space savings will vary depending on the specific attributes of the data itself.

The table below illustrates the relationship between the effective to usable and effective to raw ratios for the F910, F900, F810, F600, F200, H700, H7000, H5600, A300, and A3000 platforms:

Table 6. Effective to usable and raw relationships in various PowerScale configurations

Cluster minimum spec	Raw (TB)	Usable (TB)	Effective (TB)	Effective to Usable	Effective to Raw
F900 with 1.92TB NVMe	138	110	220	2.0:1	1.6:1
F900 with 3.84TB NVMe	276	221	442	2.0:1	1.6:1
F900 with 7.68TB NVMe	553	442	884	2.0:1	1.6:1
F900 with 15.36TB NVMe	1106	885	1770	2.0:1	1.6:1
F810 with 3.8TB SSD	228	182	365	2.0:1	1.6:1
F810 with 7.6TB SSD	456	365	730	2.0:1	1.6:1
F810 with 15.4TB SSD	924	739	1,478	2.0:1	1.6:1
F600 with 1.92TB NVMe	46	37	74	2.0:1	1.6:1
F600 with 3.84TB NVMe	92	74	148	2.0:1	1.6:1
F600 with 7.68TB NVMe	184	147	294	2.0:1	1.6:1
F200 with 0.96TB SSD	11.5	9	18	2.0:1	1.6:1
F200 with 1.92TB SSD	23	18	36	2.0:1	1.6:1
F200 with 3.84TB SSD	46	37	74	2.0:1	1.6:1
H700 with 16TB HDD	960	768	1,536	2.0:1	1.6:1
H7000 with 16TB HDD	1280	1024	2048	2.0:1	1.6:1
H5600 with 10TB HDD	800	640	1,280	2.0:1	1.6:1
H5600 with 12TB HDD	960	768	1,536	2.0:1	1.6:1
A300 with 16TB HDD	960	768	1,536	2.0:1	1.6:1
A3000 with 16TB HDD	1280	1024	2048	2.0:1	1.6:1

- Effective usable capacity assumes 1.6:1 compression ratio from raw.
- Usable capacity assumes 20% protection overhead from raw.
- F810/H700/H7000/H5600/A300/A3000 assumes 4-node chassis, F900/F600/F200 assumes 3-node cluster

The following table provides descriptions for the various OneFS reporting metrics, such as those returned by the ‘isi statistics data-reduction’ command described below. The table attempts, where appropriate, to equate the OneFS nomenclature with more general industry terminology:

Table 7. OneFS data reduction reporting metrics

Data Metric	Also Known As	Description
Protected logical	Application logical	Data size including sparse data, zero block eliminated data, and CloudPools data stubbed to a cloud tier.
Logical data	Effective Filesystem logical	Data size, excluding protection overhead and sparse data, and including data efficiency savings (compression and deduplication).
Zero-removal saved		Capacity savings from zero removal.
Dedupe saved		Capacity savings from deduplication.
Compression saved		Capacity savings from in-line compression.
Preprotected physical	Usable Application physical	Data size excluding protection overhead and excluding data efficiency savings (compression and deduplication).
Protection overhead		Size of erasure coding used to protect data.
Protected physical	Raw Filesystem physical	Total footprint of data including protection overhead FEC erasure coding) and excluding data efficiency savings (compression and deduplication).
Zero removal ratio		Efficiency ratio from zero block removal.
Dedupe ratio		Estimated deduplication ratio. Will be displayed as 1.0:1 if there are no deduplicated blocks on the cluster.
Compression ratio	Effective to Usable	Usable efficiency ratio from compression, calculated by dividing 'logical data' (green) by 'unprotected physical' (blue) and expressed as x:1
Inlined data ratio		Efficiency ratio from inlining small file data within its 8KB inode.
Data reduction ratio		Usable efficiency ratio from compression and deduplication. Will display the same value as the compression ratio if there is no deduplication on the cluster.
Efficiency ratio	Effective to Raw	Overall raw efficiency ratio, calculated by dividing 'logical data' (green) by 'protected physical' (grey) and expressed as x:1

The color scheme in this table is used throughout this paper to categorize and distinguish between the various data metrics.

The interrelation of the data capacity metrics described above can be illustrated in a graphical representation.

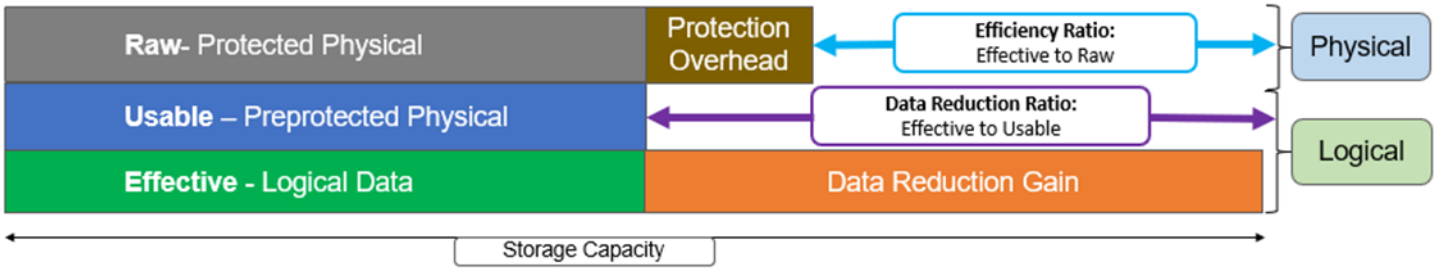

Figure 12. OneFS data capacity metrics interrelation

As we can see, the preprotected physical (usable) value, is derived by subtracting the protection overhead from the protected physical (raw) metric. Similarly, the difference in size between preprotected physical (usable) and logical data (effective) is the efficiency savings. If OneFS SmartDedupe is also licensed and running on the cluster, this data reduction savings value will reflect a combination of compression, inline deduplication, and post-process deduplication savings.

Inline data reduction efficiency reporting

Inline data efficiency statistics

OneFS provides six principal reporting methods for obtaining efficiency information with inline data reduction.

- Using the ‘isi statistics data-reduction’ CLI command
- Via the ‘isi compression’ CLI command
- Via the ‘isi dedupe’ CLI command and WebUI chart
- From the ‘isi get -O’ CLI command
- Configuring SmartQuotas reporting
- Via the ‘isi status’ CLI command
- OneFS WebUI Cluster Dashboard Storage Efficiency Summary

Isi statistics data-reduction command

The most comprehensive of the data reduction reporting CLI utilities is the ‘isi statistics data-reduction’ command. For example:

```
# isi statistics data-reduction
                      Recent Writes Cluster Data Reduction
                           (5 mins)
---------------------- ------------- ----------------------
Logical data                 6.18M                   6.02T
Zero-removal saved               0                       -
Deduplication saved         56.00k                   3.65T
Compression saved            4.16M                   1.96G
Preprotected physical        1.96M                   2.37T
Protection overhead          5.86M                 910.76G
Protected physical           7.82M                   3.40T
Zero removal ratio       1.00 : 1                       -
Deduplication ratio      1.01 : 1                2.54 : 1
Compression ratio        3.12 : 1                1.02 : 1
Data reduction ratio     3.15 : 1                2.54 : 1
Efficiency ratio         0.79 : 1                1.77 : 1
---------------------- ------------- ----------------------
```

‘Recent writes’ data to the left of the output provides precise statistics for the five-minute period before running the command. By contrast, the ‘cluster data reduction’ metrics on the right of the output are slightly less than real time but reflect the overall data and efficiencies across the cluster.

Note: In OneFS 9.1 and earlier, the right-side column metrics are designated by the ‘Est’ prefix, denoting an estimated value. However, in OneFS 9.2 and later, the ‘logical data’ and ‘preprotected physical’ metrics are now tracked and reported accurately, rather than estimated.

The ratio data in each column is calculated from the values above it. For instance, to calculate the data reduction ratio, the ‘logical data’ (effective) is divided by the ‘preprotected physical’ (usable) value. From the output above, the ratio would be as follows:

6.02 / 2.37 = 2.54 Or a **Data Reduction** ratio of **2.54:1**

Similarly, the 'efficiency ratio' is calculated by dividing the 'logical data' (effective) by the 'protected physical' (raw) value. From the output above, this yields:

6.02 / 3.40= 1.77 Or an **Efficiency** ratio of **1.77:1**

Isi compression stats command

From the OneFS CLI, the 'isi compression stats' command provides the option to either view or list compression statistics. When run in 'view' mode, the command returns the compression ratio for both compressed and all writes, plus the percentage of incompressible writes, for a prior five-minute (300 seconds) interval. For example:

```
# isi compression stats view
  stats for 300 seconds at: 2021-04-14 15:46:04 (1618429564)
    compression ratio for compressed writes:  3.12 : 1
    compression ratio for all writes:  3.12 : 1
    incompressible data percent:  6.25%
    total logical blocks: 784
    total physical blocks: 251
    writes for which compression was not attempted:  0.00%
```

If the 'incompressible data' percentage is high in a mixed cluster, there is a strong likelihood that the majority of the writes are going to a non-compression pool.

The 'isi compression stats' CLI command also accepts the 'list' argument, which consolidates a series of recent reports into a list of the compression activity across the file system. For example:

```
# isi compression stats list
Statistic   compression  overall  incompressible  logical   physical  compression
            ratio        ratio    %               blocks    blocks    skip %
1618425636  3.07:1       3.07:1   10.59%          68598     22849     1.05%
1618425636  3.20:1       3.20:1   7.73%           4142      1293      0.00%
1618425636  3.14:1       3.14:1   8.24%           352       112       0.00%
1618425636  2.90:1       2.90:1   9.60%           354       122       0.00%
1618425636  1.29:1       1.29:1   75.23%          10839207  8402380   0.00%
```

The 'isi compression stats' data is used for calculating the right-side estimated 'Cluster Data Reduction' values in the 'isi statistics data-reduction' command described above. It also provides a count of logical and physical blocks and compression ratios, plus the percentage metrics for incompressible and skipped blocks.

The value in the 'statistic' column at the left of the table represents the epoch timestamp for each sample. This epoch value can be converted to a human readable form using the 'date' CLI command. For example:

```
# date -d 1618425636
Wed Apr 14 15:47:34 EDT 2021
```

Isi dedupe stats command and WebUI chart

From the OneFS CLI, the 'isi dedupe stats' command provides cluster deduplication data usage and savings statistics, in both logical and physical terms. For example:

```
# isi dedupe stats
      Cluster Physical Size: 86.14T
          Cluster Used Size: 3.43T
  Logical Size Deduplicated: 4.01T
             Logical Saving: 3.65T
Estimated Size Deduplicated: 5.42T
  Estimated Physical Saving: 4.93T
```

Inline deduplication and post-process SmartDedupe both deliver very similar end results, just at different stages of data ingestion. Since both features use the same core components, the results are combined. As such, the isi deduplication stats output reflects the sum of both inline deduplication and SmartDedupe efficiency. Similarly, the OneFS WebUI's deduplication savings histogram combines the efficiency savings from both inline deduplication and SmartDedupe.

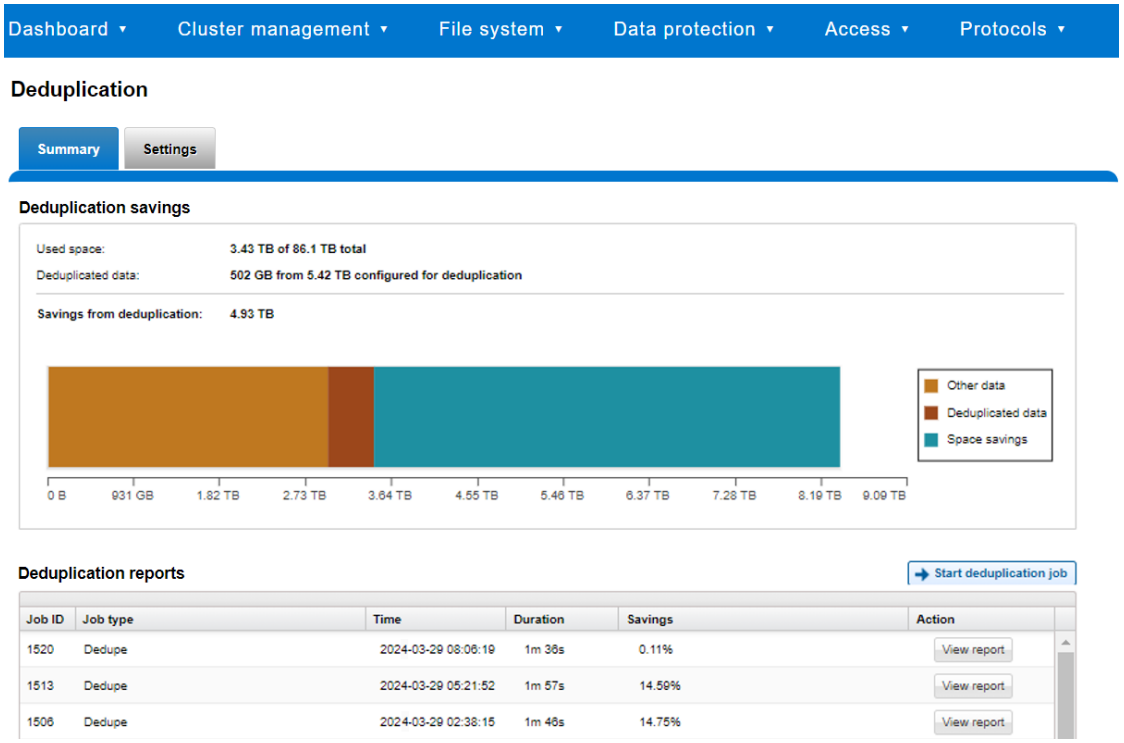

Figure 13. Deduplication cluster capacity savings WebUI chart

The deduplication statistics do not include zero block removal savings. Since zero block removal is technically not due to data deduplication, it is tracked separately but is included as part of the overall data reduction ratio.

Isi get statistics

OneFS includes a '-O' logical overlay flag to 'isi get' CLI utility for viewing a file's compression details.

For example:

```
# isi get -DDO file1
* Size:           167772160
* PhysicalBlocks: 10314
```

```
* LogicalSize:    167772160
PROTECTION GROUPS
lbn0: 6+2/2
2,11,589365248:8192[COMPRESSED]#6
0,0,0:8192[COMPRESSED]#10
2,4,691601408:8192[COMPRESSED]#6
0,0,0:8192[COMPRESSED]#10
Metatree logical blocks:
zero=32 shadow=0 ditto=0 prealloc=0 block=0 compressed=64000
```

The logical overlay information is described under the 'protection groups' output. This example shows a compressed file where the sixteen-block chunk is compressed down to six physical blocks (#6) and ten sparse blocks (#10). Under the 'Metatree logical blocks' section, a breakdown of the block types and their respective quantities in the file is displayed - including a count of compressed blocks.

When compression has occurred, the 'df' CLI command will report a reduction in used disk space and an increase in available space. The 'du' CLI command will also report less disk space used.

A file that for whatever reason cannot be compressed will be reported as such:

```
4,6,900382720:8192[INCOMPRESSIBLE]#1
```

OneFS 9.2 and later releases use inode version 8, which includes a couple of additional inode delta attributes for storing data reduction metrics. These new attributes are displayed by the 'isi get -D' CLI command, and report a file's physical data blocks, compressed size, and protection blocks. For example:

```
# isi get -D file1
POLICY   W   LEVEL PERFORMANCE COAL  ENCODING      FILE
IADDRS
default      6+2/2 concurrency on    UTF-8         file1
<1,4,201744384:8192>, <2,3,59752448:8192>, <4,3,176726016:8192>
ct: 1613083429 rt: 0
************************************************
* IFS inode: [ 1,4,201744384:8192, 2,3,59752448:8192,
4,3,176726016:8192 ]
************************************************
*
*  Inode Version:      8
*  Dir Version:        2
*  Inode Revision:     214
*  Inode Mirror Count: 3
*  Recovered Flag:     0
*  Restripe State:     0
*  Link Count:         1
*  Size:               524288000
*  Mode:               0100644
*  Flags:              0xe0
*  SmartLinked:        False
*  Physical Blocks:    15552
```

```
*   Phys. Data Blocks:   9299
*   Compressed Size:     20.528%
*   Protection Blocks:   6064
```

SmartQuotas data reduction efficiency reporting

OneFS SmartQuotas reports the capacity saving from inline data reduction as a storage efficiency ratio. SmartQuotas reports efficiency as a ratio across the chosen dataset as specified in the quota path field. The efficiency ratio is for the full quota directory and its contents, including any overhead, and reflects the net efficiency of compression and deduplication. On a cluster with licensed and configured SmartQuotas, this efficiency ratio can be easily viewed from the WebUI by going to File System > SmartQuotas > Quotas and Usage. In OneFS 9.2 and later, in addition to the storage efficiency ratio, the data reduction ratio is also displayed.

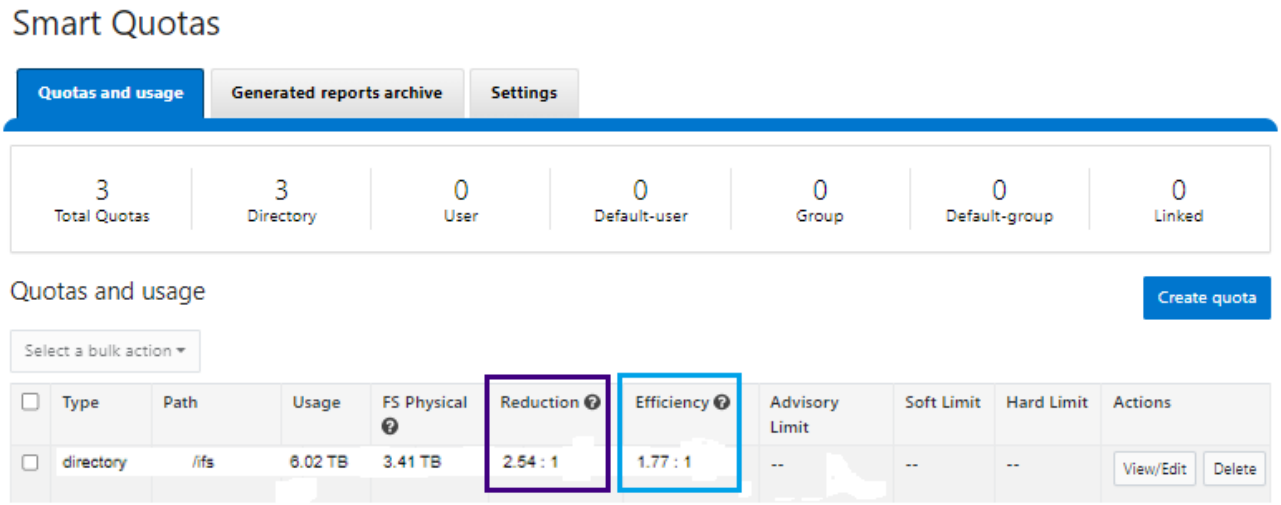

Figure 14. OneFS WebUI SmartQuotas Quotas and Usage Status Detailing Data Reduction and Efficiency Ratios

Similarly, the same data can be accessed from the OneFS command line using is ‘isi quota quotas list’ CLI command. For example:

```
 # isi quota quotas list
Type      AppliesTo  Path   Snap  Hard  Soft  Adv  Used   Reduction   Efficiency
-------------------------------------------------------------------------------
directory DEFAULT    /ifs   No    -     -     -    6.02T  2.54 : 1    1.77 : 1
-------------------------------------------------------------------------------
Total: 1
```

More detail, including both the physical (raw) and logical (effective) data More detail, including both the physical (raw) and logical (effective) data capacities, is also available using the ‘isi quota quotas view <path> <type>’ CLI command. For example:

```
# isi quota quotas view /ifs directory
                        Path: /ifs
                        Type: directory
                   Snapshots: No
                    Enforced: No
                   Container: No
                      Linked: No
                       Usage
                               Files: 5759676
        Physical(With Overhead): 6.93T
       FSPhysical(Deduplicated): 3.41T
        FSLogical(W/O Overhead): 6.02T
       AppLogical(ApparentSize): 6.01T
                  ShadowLogical: -
                   PhysicalData: 2.01T
                     Protection: 781.34G
    Reduction(Logical/Data): 2.54 : 1
Efficiency(Logical/Physical): 1.77 : 1
```

To configure SmartQuotas for inline data efficiency reporting, create a directory quota at the top-level file system directory of interest, for example /ifs. Creating and configuring a directory quota is a simple procedure and can be performed from the WebUI, as follows:

Go to **File System > SmartQuotas > Quotas and Usage**, and select **Create a Quota**. In the create pane, field, set the Quota type to **Directory quota**, add the preferred top-level path to report on, select **File system logical size** for **Quota Accounting**, and set the **Quota Limits** to **Track storage without specifying a storage limit**. Finally, select the **Create Quota** button to confirm the configuration and activate the new directory quota.

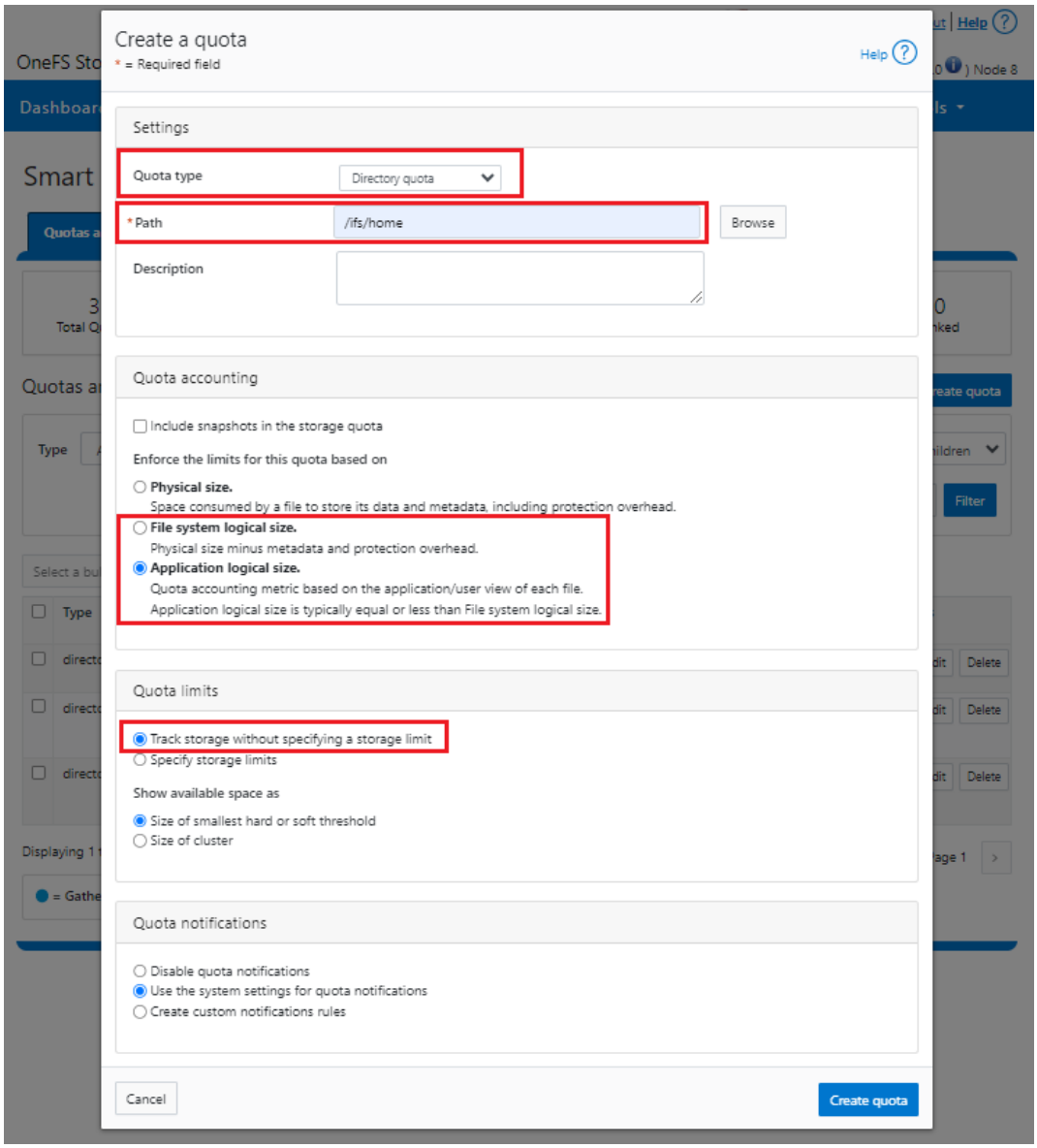

Figure 15. OneFS WebUI SmartQuotas Directory Quota configuration

The efficiency ratio is a single, current-in time efficiency metric that is calculated per quota directory and includes the sum of inline compression, zero block removal, inline deduplication and SmartDedupe. This is in contrast to a history of stats over time, as reported in the 'isi statistics data-reduction' CLI command output, described above. As such, the efficiency ratio for the entire quota directory will reflect what is actually there.

The quota directory efficiency ratio, and other statistics, are not available using the platform API as of OneFS 9.0.

Isi status command

The isi status CLI command output includes a Data Reduction field:

```
# isi status
Cluster Name: f8101
Cluster Health:     [  OK ]
Data Reduction:     2.54 : 1
Storage Efficiency: 1.77 : 1
Cluster Storage:  HDD                 SSD Storage
Size:             0 (0 Raw)           82.9T (86.1T Raw)
VHS Size:         3.2T
Used:             0 (n/a)             3.4T (4%)
Avail:            0 (n/a)             79.5T (96%)

                   Health  Throughput (bps)  HDD Storage      SSD Storage
ID |IP Address     |DASR |  In   Out   Total| Used / Size     |Used / Size
---+---------------+-----+-----+-----+-----+-----------------+----------------
  1|10.245.110.69  | OK  |    0|    0|    0|(No Storage HDDs)| 878G/20.7T(  4%)
  2|10.245.110.70  | OK  |    0|73.9k|73.9k|(No Storage HDDs)| 879G/20.7T(  4%)
  3|10.245.110.71  | OK  |    0| 149k| 149k|(No Storage HDDs)| 879G/20.7T(  4%)
  4|10.245.110.72  | OK  |    0| 494k| 494k|(No Storage HDDs)| 879G/20.7T(  4%)
---+---------------+-----+-----+-----+-----+-----------------+----------------
Cluster Totals:          |    0| 717k| 717k|    0/    0( n/a)| 3.4T/82.9T(  4%)

     Health Fields: D = Down, A = Attention, S = Smartfailed, R = Read-Only
```

OneFS WebUI cluster dashboard storage efficiency summary

The OneFS WebUI cluster dashboard displays a storage efficiency tile. This tile shows physical and logical space utilization histograms and reports the capacity saving from inline data reduction as a storage efficiency ratio. In OneFS 9.2 and later, a data reduction ratio is also included in the dashboard view. This cluster status view is displayed by default upon opening the OneFS WebUI in a browser and can be easily accessed by going to **File System > Dashboard > Cluster Overview**.

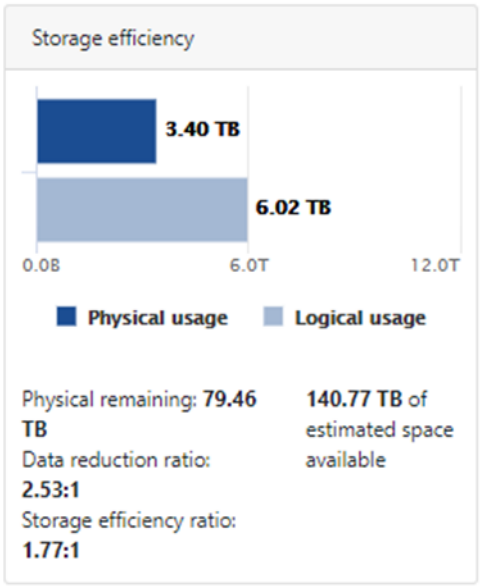

Figure 16. OneFS WebUI cluster status dashboard – Storage efficiency summary tile

All above storage efficiency tools are available on any cluster running OneFS 8.2.1 and later. However, the inline compression metrics will only be relevant for clusters containing compression node pools.

Performance with inline data reduction

As with most things in life, data efficiency is a compromise. To gain increased levels of storage efficiency, additional cluster resources (CPU, memory, and disk IO) are utilized to perform the compressing and deduping and re-inflating of files. As such, the following factors can affect the performance of inline data reduction and the I/O performance of compressed and deduplicated pools:

- Application and the type of dataset being used
- Data access pattern (for example, sequential compared to random access, the size of the I/O)
- Compressibility and duplicity of the data
- Amount of total data
- Average file size
- Nature of the data layout
- Hardware platform: the amount of CPU, RAM, and type of storage in the system
- Amount of load on the system
- Level of protection

Clearly, hardware offload compression will perform better, both in terms of speed and efficiency, than the software fallback option. This improvement is evident on both on F810 nodes where the hardware compression engine has been disabled, and on all other node types where software data reduction is the only available option.

Another important performance impact consideration with inline data efficiency is the potential for data fragmentation. After compression or deduplication, files that previously enjoyed contiguous on-disk layout will often have chunks spread across less optimal file system regions. This can lead to slightly increased latencies when accessing these files directly from disk, rather than from cache.

Because inline data reduction is a data efficiency feature rather than performance-enhancing tool, usually the consideration will be around cluster impact management. This consideration includes both the client data access performance front and from the data reduction execution perspective, as additional cluster resources are consumed when shrinking and inflating files.

With inline data reduction enabled, highly incompressible datasets may experience a small performance penalty. Conversely, for highly compressible and duplicate data, there may be a performance boost. Workloads performing small, random operations will likely see a small performance degradation.

Since they reside on the same card, the compression FPGA engine shares PCIe bandwidth with the node’s backend Ethernet interfaces. In general, there is plenty of bandwidth available. However, a best practice is to run incompressible performance

streaming workflows on F810 nodes with inline data reduction disabled to avoid any potential bandwidth limits.

In general, rehydration requires considerably less overhead than compression.

When considering effective usable space on a cluster with inline data reduction enabled, understand that every capacity saving from file compression and deduplication also serves to reduce the per-TB compute ratio (CPU, memory). For performance workloads, the recommendation is to size for performance (IOPS, throughput) rather than effective capacity.

Similarly, it is challenging to broadly characterize the inline deduplication performance overhead with any accuracy since it depends on various factors including the duplicity of the dataset, whether matches are found against other LINs or SINs. Workloads requiring a large amount of deduplication might see an impact of 5-10%, although they experience an attractive efficiency ratio. In contrast, certain other workloads may see a slight performance gain because of inline deduplication. If there is block scanning but no deduplication to perform, the overhead is typically in the 1-2% range.

Typically, SmartDedupe space savings in addition to inline deduplication fail a cost benefit analysis against performance trade-offs. Minimally intrusive Cost-Benefit analysis can be performed in the following manner:

- With only inline deduplication in operation, obtain a performance baseline of user sensitive Key Performance Indicators (KPIs).
- While monitoring the chosen KPIs, run the DedupeAssessment job to obtain an estimate of additional space savings.

Note: This job puts less load on the cluster than the regular SmartDedupe job.

- Review estimated space saving vs performance changes but be aware that SmartDedupe will add more load than observed during the DedupeAssessment job run.
- Enable SmartDedupe if estimated space savings offer business benefit above performance trade-off.

Inline data reduction licensing

Inline data reduction is included as a core component of OneFS on the F910, F900, F810, F710, F600, F210, F200, H700/7000, H5600, and A300/3000 hardware platforms and does not require a product license key to activate. Inline compression is enabled by default, and inline deduplication can be activated using the following CLI command:

```
# isi dedupe inline settings modify --enabled=True
```

Note that an active SmartQuotas license is required to use quota reporting. A SmartQuotas license key can be purchased through your Dell account team. An unlicensed cluster will show a SmartQuotas warning until a valid product license has been purchased and applied to the cluster.

License keys can be easily added using the 'Activate License' section of the OneFS WebUI, accessed by going to Cluster Management > Licensing.

Inline data reduction and workflows

Overview

Below are some examples of typical space reclamation levels that have been achieved with OneFS inline data reduction efficiency.

These data efficiency space savings values are provided solely as rough guidance. Since no two datasets are alike (unless they are replicated), actual results can and will vary considerably from these examples.

Table 8. Typical workload space savings with inline data reduction

Workflow or data type	Typical efficiency ratio	Typical space savings
Home directories or file shares	1.3:1	25%
Engineering source code	1.4:1	30%
EDA data	2:1	50%
Genomics data	2.2:1	55%
Oil and gas	1.4:1	30%
Pre-compressed data	N/A	No savings

Tests of various datasets have demonstrated that data efficiency ratios can easily range from 1:1 (no reduction) to over 3:1.

Inline compression estimation with Live Optics Dossier

The Live Optics Dossier utility can be used to estimate the potential benefits of OneFS' inline data reduction on a dataset. Dossier is available for Windows and Linux and has no dependency on a Dell PowerScale cluster. This makes it useful for analyzing and estimating efficiency across real data in place, without the need for copying data onto a cluster.

Dossier operates in three phases:

Table 9. Live Optics Dossier phases

Dossier phase	Description
Discovery	Users manually browse and select root folders on the local host to analyze.
Collection	Once the paths to folders have been selected, Dossier will begin walking the file system trees for the target folders. This process will likely take up to several hours for large file systems. Walking the file system has a similar impact to a malware or anti-virus scan in terms of the CPU, memory, and disk resources that will be utilized during the collection. Customizable options allow the user to deselect more invasive operations and govern the CPU and memory resources allocated to the Dossier collector.
Reporting	Users upload the resulting '.dossier' file to create a Microsoft PowerPoint and Excel report.

To obtain a Live Optics Dossier report, first download, extract, and run the Live Optics collector (from https://www.liveoptics.com/). Local and remote UNC paths can be added for scanning. Ensure you are authenticated to the wanted UNC path before adding it to Dossier's 'custom paths' configuration.

Be aware that the Dossier compression option only processes the first 64 KB of each file to determine its compressibility. Also, the configuration samples 20% (by default) of the dataset, but this setting is configurable. Increasing this value improves the accuracy of the estimation report, albeit at the expense of extended job execution time.

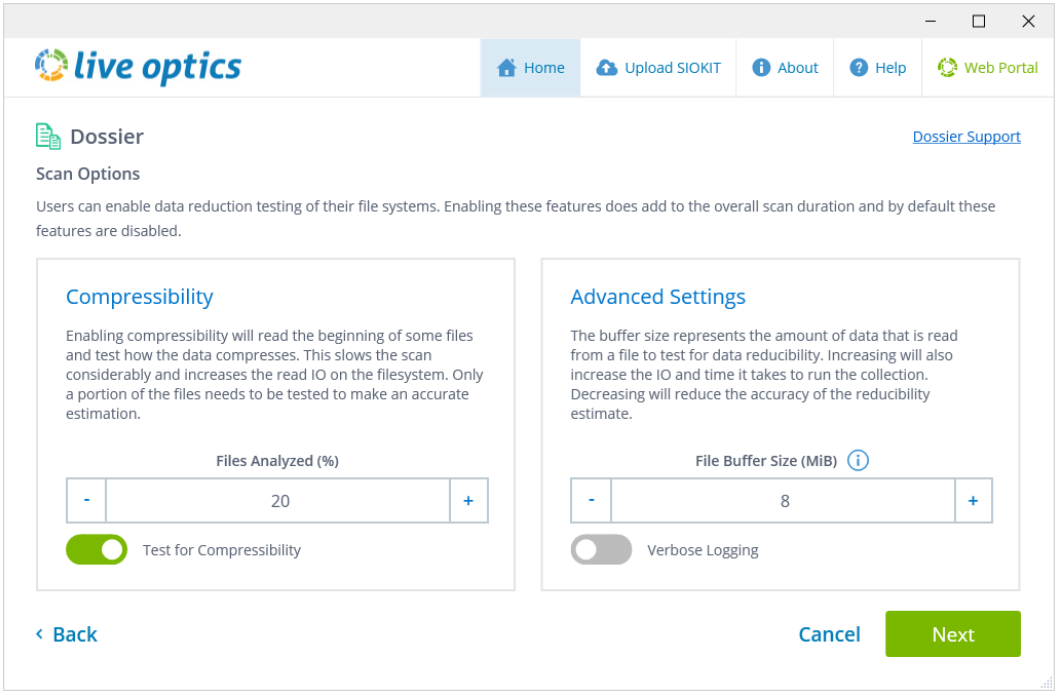

Figure 17. Dossier configuration

The compressibility scan runs rapidly, with minimal CPU and memory resource consumption. It also provides thread and memory usage controls, progress reporting, and a scheduling option to allow throttling of scanning during heavy usage windows.

When the scan is complete, a '*.dossier' file is generated. This file must be manually uploaded to the Live Optics web portal to generate the PowerPoint and Excel report. See https://support.liveoptics.com/hc/en-us/articles/17358092010779 for more information on uploading the .dossier file to generate the reports.

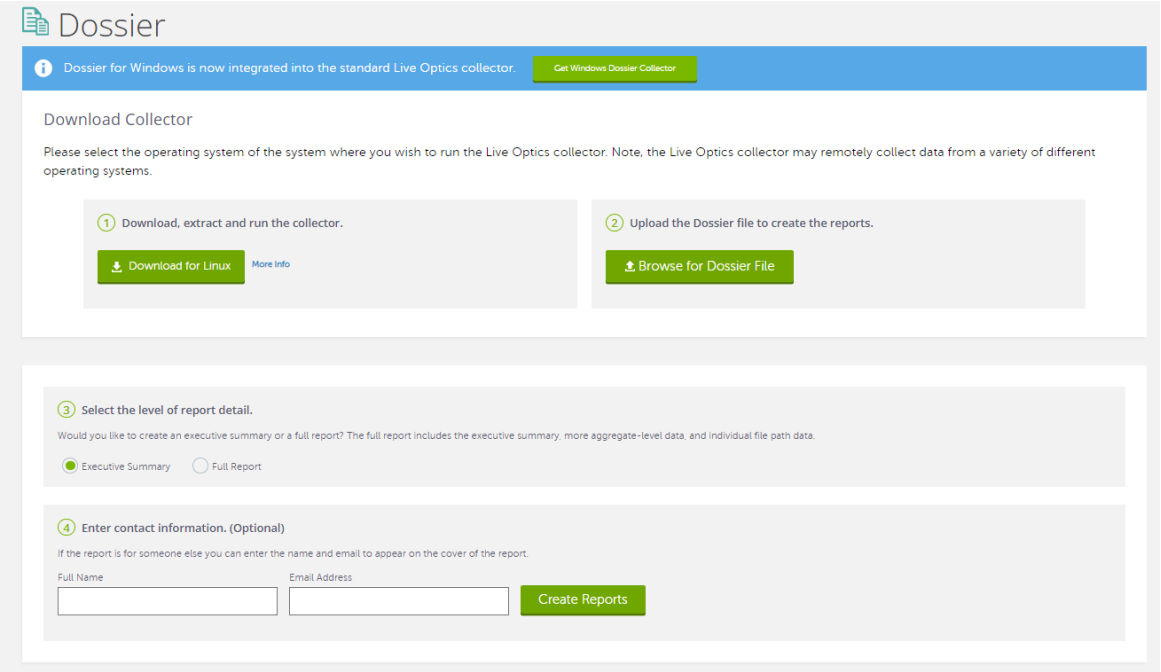

Figure 18. Live Optics Dossier file upload

Once uploaded and processed, a PowerPoint and Excel report are generated in real time and downloaded to your computer.

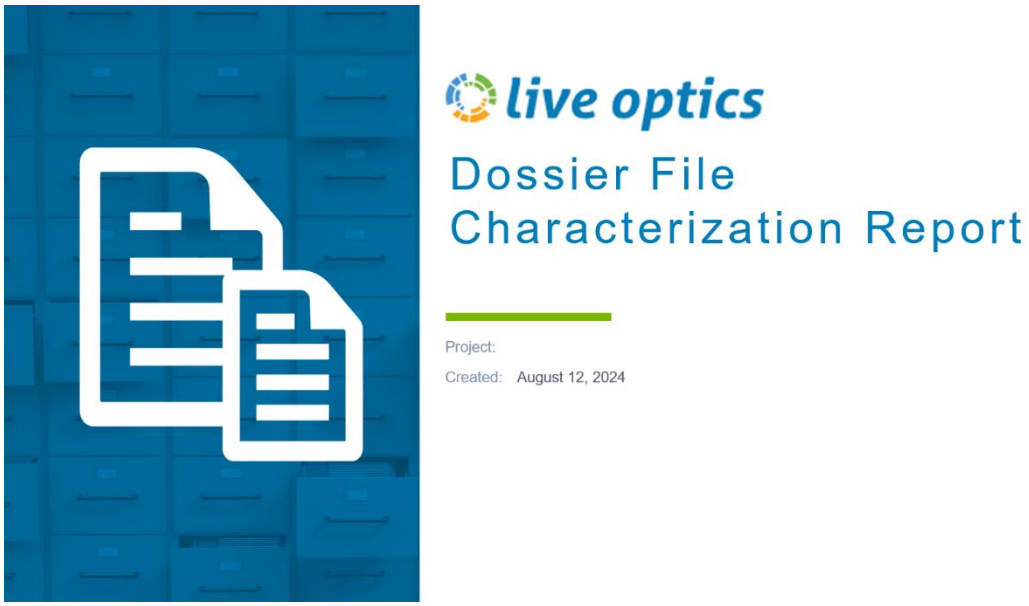

Figure 19. Dossier PowerPoint report

Compression reports are easy to comprehend. If multiple SMB shares or paths are scanned, a summary is generated at the beginning of the report, followed by the details of each individually selected path.

Live Optics Dossier can be found at: https://app.liveoptics.com/tools/dossier

Documentation is at: https://support.liveoptics.com/hc/en-us/sections/360012122533

When running the Live Optics Dossier tool, keep in mind the following considerations:

- Does not provide exactly the same algorithm as the OneFS hardware inline compression.
- Dossier looks at the software compression, not the hardware compression. So actual results will generally be better than Dossier report.
- There will be some data for which dossier overestimates compression, for example with files whose first blocks are significantly more compressible than later blocks.
- Intended to be run against any SMB shares on any storage array or DAS. No NFS export support.
- Dossier tool can take a significant amount of time to run against a large dataset.
- By default, it only samples a portion (first 64 KB) of the data, so results can be inaccurate.
- Dossier does not attempt to compress files with certain known extensions which are generally uncompressible.
- Dossier assessment tool only provides the size of the uncompressed and compressed data. It does not provide performance estimates of different compression algorithms.

Inline deduplication efficiency estimation

A dry run Dedupe Assessment job is provided to help estimate the amount of space savings that will be seen on a dataset. Run against a specific directory or set of directories on a cluster, the deduplication assessment job reports a total potential space savings. The assessment job uses a separate configuration. It also does not require a product license and can be run before purchasing F910, F900, F810, F600, F200, F700/7000, H5600, and A300/3000 hardware to determine whether deduplication is appropriate for a particular dataset or environment.

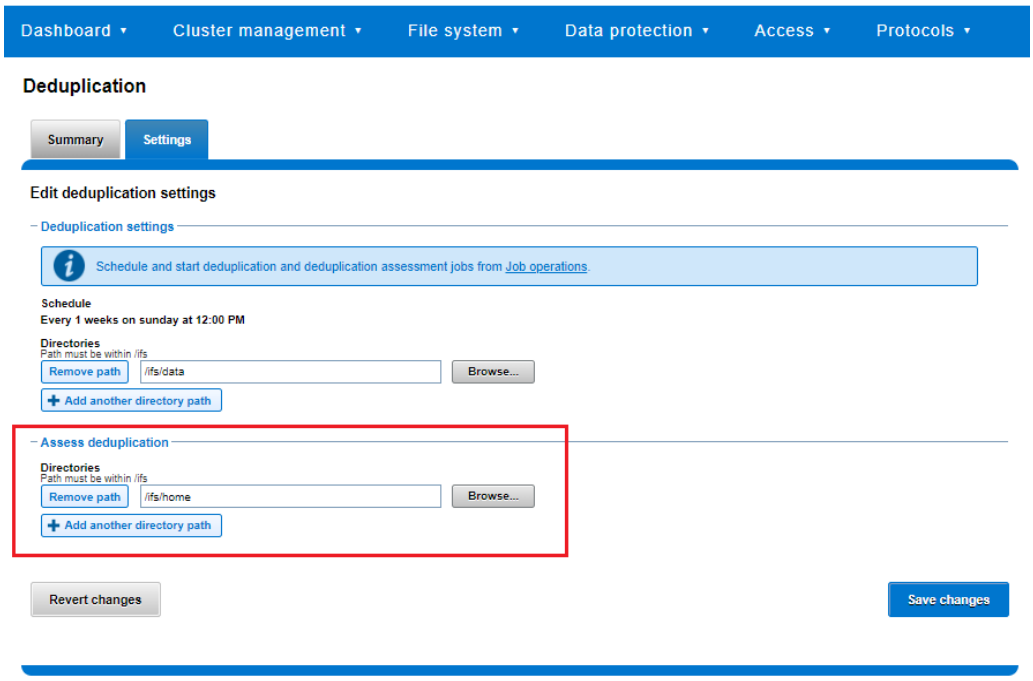

Figure 20. Deduplication assessment job configuration

The deduplication assessment job uses a separate index table to both inline deduplication and SmartDedupe. For efficiency, the assessment job also samples fewer candidate blocks and does not actually perform deduplication. Using the sampling and consolidation statistics, the job provides a report which estimates the total deduplication space savings in bytes.

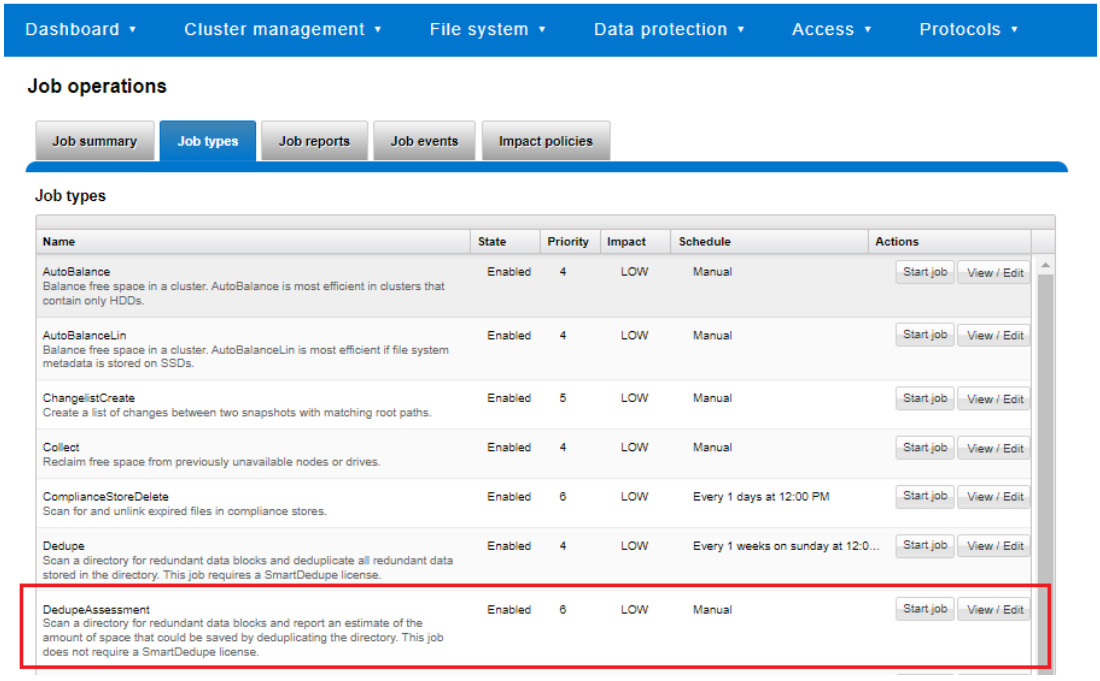

Figure 21. DedupeAssessment job control using the OneFS WebUI

The DedupeAssessment job can also be run from the OneFS command line (CLI):

```
# isi job jobs start DedupeAssessment
```

Alternatively, inline deduplication can be enabled in assessment mode:

```
# isi dedupe inline settings modify –mode assess
```

One the job has completed, review the following three metrics from each node:

```
# sysctl efs.sfm.inline_dedupe.stats.zero_block
# sysctl efs.sfm.inline_dedupe.stats.dedupe_block
# sysctl efs.sfm.inline_dedupe.stats.write_block
```

The formula to calculate the estimated deduplication rate from these statistics is:

dedupe_block / write_block * 100 = dedupe%

The deduplication assessment does not differentiate the case of a fresh run from the case where a previous SmartDedupe job has already performed some sharing on the files in that directory. We recommend that the user should run the assessment job once on a specific directory, since it does not provide incremental differences between instances of the job.

Inline data reduction and OneFS feature integration

The following table describes the integration, influence, and compatibility between inline data reduction and the various OneFS data services.

Except for the job engine and non-disruptive upgrade (NDU), the following services each require a product license and are not enabled, configured, and active by default on a cluster.

Table 10. OneFS data reduction and data services integration

OneFS feature	Detail
SyncIQ	If compressed or deduplicated (or both) data is replicated to a target cluster with SyncIQ, those files are automatically decompressed and rehydrated on read and transferred and written to the target in their uncompressed form. However, if the target happened to also be a compression node pool, inline data reduction would occur.
NDMP Backup	Because files are backed up as if the files were not compressed or deduplicated, backup and replication operations are not faster for compressed or deduplicated data. OneFS NDMP backup data will not be compressed unless compression is provided by the backup vendor's DMA software. However, compression is often provided natively by the backup tape or VTL device.
SnapshotIQ	Compression will not affect the data stored in a snapshot. However, snapshots can be created of compressed data. If a data reduction tier is added to a cluster that already has a significant amount of data stored in snapshots, it will take time before the snapshot data is affected by compression. Newly created snapshots will contain compressed data, but older snapshots will not. Deduplicated data can also end up in a snapshot if the HEAD file is deduplicated and then the shadow references are COWed to the snapshot. While OneFS inline compression works with writable snapshots data, deduplication is not supported, and existing files under writable snapshots will be ignored by inline deduplication. However, inline deduplication can occur on any new files created fresh on the writable snapshot.

OneFS feature	Detail
SmartLock	Inline data reduction is compatible with SmartLock, OneFS' data retention and compliance product. Compression and deduplication deliver storage efficiency for immutable archives and write once, read many (or WORM) protected datasets. The F910, F900, F810, F600, F200, F700/7000, H5600, and A300/3000 hardware all support compressing and deduping data in files from a SmartLock/WORM domain, including compressing existing files that are currently stored in an uncompressed state. Since the logical content of the file data is unchanged, WORM compliance is unaffected.
SED Encryption	Encryption with SED drives is supported on clusters with F810 nodes running OneFS 8.2.1 (15.4 TB SSD drives only), H5600 nodes running OneFS 8.2.2, F600 & F200 nodes running OneFS 9.0, F900 nodes running OneFS 9.2, F710 and F210 nodes running OneFS 9.7, and F910 nodes running OneFS 9.8.
SmartQuotas	OneFS SmartQuotas is one of the principal methods for inline data reduction efficiency reporting. Quotas account for compressed files as if they consumed both shared and unshared data. From the quota side, compressed files appear no differently than regular files to standard quota policies.
SmartPools	Compressed files will only reside on compression nodes and will not span SmartPools node pools or tiers. This is to avoid potential performance or protection asymmetry which could occur if portions of a file live on different classes of storage. SmartPooled data will be uncompressed before it is moved, so full uncompressed capacity will be required on the compressed pool.
CloudPools	Although CloudPools can use compression to transfer data to the service provider, in OneFS 8.2.1 and later, compressed or deduplicated data cannot be exported directly from disk without incurring a decompression or compression cycle. CloudPools sees uncompressed data and then re-compresses data. CloudPools uses a different chunk size that inline compression.
Non-disruptive Upgrade	Inline data reduction is only available on F810 nodes in OneFS 8.2.1, H5600 nodes in OneFS 8.2.2, F600 and F200 nodes in OneFS 9.0, F900 nodes in OneFS 9.2, H700/7000 and A300/3000 nodes in OneFS 9.2.1, F710 and F210 nodes in OneFS 9.7, and F910 nodes in OneFS 9.8. Gen6 clusters with an Ethernet back end, running earlier versions of OneFS 8.x code can be non-disruptively upgraded to OneFS 8.2.1 and later.
File Clones	File cloning places data in shadow stores and notifies SmartDedupe by identifying the inode of the cloned LIN so that SmartDedupe samples the shadow references (normally it skips them).

<table>
<tr><th>OneFS feature</th><th>Detail</th></tr>
<tr><td>SmartDedupe</td><td>SmartDedupe post-process deduplication is compatible with inline data reduction and conversely. Inline compression is able to compress OneFS shadow stores. However, for SmartDedupe to process compressed data, the SmartDedupe job will have to decompress it first in order to perform deduplication, which is an addition resource overhead.

Currently neither SmartDedupe nor inline deduplication are immediately aware of the duplicate matches that each other finds. Both inline deduplication and SmartDedupe could deduplicate blocks containing the same data to different shadow store locations, but OneFS is unable to consolidate the shadow blocks together. When blocks are read from a shadow store into L1 cache, they are hashed and added into the in-memory index where they can be used by inline deduplication.

Unlike SmartDedupe, inline deduplication can deduplicate a run of consecutive blocks to a single block in a shadow store.
<table>
<tr><th>Inline deduplication</th><th>SmartDedupe</th></tr>
<tr><td>Globally enabled</td><td>Directory tree based</td></tr>
<tr><td>Will process small files</td><td>Skips files < 32 KB (by default)</td></tr>
<tr><td>Will deduplicate sequential runs of blocks of same data to single blocks</td><td>Can only deduplicate between files</td></tr>
<tr><td>Per-node, non-persistent in-memory index</td><td>Large persistent on-disk index</td></tr>
<tr><td>Can convert copy operations to clone</td><td>Post process only</td></tr>
<tr><td>Opportunistic</td><td>Exhaustive</td></tr>
</table>
</td></tr>
<tr><td>Small File Storage Efficiency (SFSE)</td><td>SFSE is mutually exclusive to all the other shadow store consumers (file clones, inline deduplication, SmartDedupe). Files can either be packed with SFSE, or cloned or deduplicated, but not both.

Inlined files (small files with their data stored in the inode) will not be deduplicated and non-inlined datafile that are once deduplicated will not inline afterwards.</td></tr>
<tr><td>Job Engine</td><td>Only the jobs which access logical data will incur compression or decompression (or both) overhead costs. These include:

SmartPools, when moving data to or from a compressed node pool.

IntegrityScan, when working on compressed data.

FlexProtect, in the event of spillover to another nodepool.

SmartDedupe must decompress data first to perform deduplication, which is an addition resource expense.

Other jobs working on metadata and physical data will be unaffected by inline data reduction.</td></tr>
<tr><td>InsightIQ</td><td>OneFS InsightIQ (PowerScale’s multi-cluster reporting and trending analytics tool) is compatible with inline data reduction and will report efficiency savings.</td></tr>
</table>

Inline data reduction best practices

For optimal cluster performance, we recommend observing the following inline data reduction best practices. Some of this information may be covered elsewhere in this paper.

- Inline data reduction is supported on F910, F900, F810, F710, F600, F210, F200, H700/7000, H5600, A300/3000 nodepools only. Legacy F800 nodes cannot be upgraded or converted to F810 nodes.
- Run the assessment tool on a subset of the data to be compressed or deduplicated.
- When replicating compressed or deduplicated (or both) data, to avoid running out of space on target, it is important to verify that the logical data size (the amount of storage space saved plus the actual storage space consumed) does not exceed the total available space on the target cluster.
- In general, additional capacity savings may not warrant the overhead of running SmartDedupe on node pools with inline deduplication enabled. See the 'Performance with Inline Data Reduction' chapter for additional detail.
- Data reduction can be disabled on a cluster if the overhead of compression and deduplication is considered too high or performance is impacted, or both.
- The software data reduction fall-back option on F810 nodes is less performant, more resource intensive, and less efficient (lower compression ratio) that hardware data reduction. Consider removing F810 nodes with failing offload hardware from the node pool.
- Run the deduplication assessment job on a single root directory at a time. If multiple directory paths are assessed in the same job, you will not be able to determine which directory should be deduplicated.
- Recommend enabling inline deduplication before rebooting the F910, F900, F810, F600, F200, and H5600 nodes in a cluster.

Inline data reduction considerations

Overview

Inline data reduction is supported with the following caveats:

- OneFS 8.2.1and later will support from 4 to 252 F810 nodes, or 36 chassis, per cluster.
- OneFS 9.0 will support from 4 to 252 F810 or H5600 nodes, or from 3 to 252 F600 or F200 nodes per cluster.
- OneFS 9.2 will support from 4 to 252 F810 or H5600 nodes, or from 3 to 252 F900, F600, or F200 nodes per cluster.
- OneFS 9.2.1 and later will support from 4 to 252 F810, H5600, F700/7000 or A300/3000 nodes, or from 3 to 252 F900, F600, or F200 nodes per cluster.
- OneFS 9.7 and later will support from 4 to 252 F810, H5600, F700/7000 or A300/3000 nodes, or from 3 to 252 F900, F710, F600, F210, or F200 nodes per cluster.

- OneFS 9.8 and later will support from 4 to 252 F810, H5600, F700/7000 or A300/3000 nodes, or from 3 to 252 F910, F900, F710, F600, F210, or F200 nodes per cluster.
- Data reduction savings depend heavily on factors like the data, cluster composition, and protection level.
- Compressed and deduplicated data does not exit the file system compressed or deduplicated in any shape or form.
- Decompression is substantially less expensive than compression.
- Inline data reduction is exclusive to the F910, F900, F810, F710, F600, F210, F200, H700/7000, H5600, and A300/3000 platforms and does not require a software license.
- There is no compatibility or equivalency between F800 and F810 nodes: They cannot share the same node pool and the F800 nodes will not be able to store compressed data.
- There is no OneFS WebUI support for data reduction. Configuration and management are through the CLI only.
- Partial writes to compression chunks may require reading the entire compression chunk first and decompressing it. This is true even if most of the compression chunk is being written.
- Modifications to compression chunks may require rewriting the entire compression chunk even if only a single logical block is changed.
- Some workloads will have data access patterns that exacerbate the above issues and have the potential to cause more writes than if compression was not used.
- Data integrity failures with compressed data will likely mean that corruption does not just affect a single block but instead the entire compression chunk.
- If SmartPools is used on a mixed cluster containing F910, F900, F810, F710, F600, F210, F200, H700/7000, H5600, or A300/3000 nodes, data will only be compressed and/or inline deduplicated when it physically resides on the node pool or pools. If data is tiered to non-compression node pools it will be uncompressed before it is moved, so full uncompressed capacity will be required on the compressed pool. Conversely, if SmartPools moves data between compression pools, for example F810 to F200, the data will remain in a compressed state throughout the transfer.
- Post-process SmartDedupe can run in concert with compression and inline deduplication. It is supported but not widely used. The SmartDedupe job will have to decompress data first to perform deduplication, which is an additional resource expense.
- Even though compressed files are unintelligible when stored on disk, this does not satisfy the encryption requirements for secure data at rest compliance. However, PowerScale nodes are available with SED drives.
- InsightIQ is not yet fully integrated with inline data reduction and will not report compression savings. This will be addressed in a future release.

- As discussed earlier, inline compression is not free. There is always trade-off between cluster resource consumption (CPU, memory, disk), the potential for data fragmentation and the benefit of increased space efficiency.
- Since compression extends the capacity of a cluster, it also has the effect of reducing the per-TB compute resource ratio (CPU, memory, I/O).
- Depending on an application's I/O profile and the effect of Inline data reduction on the data layout, read and write performance and overall space savings can vary considerably.
- OneFS metadata structures (inodes, b-trees) are not compressed.
- Since compression trades cluster performance for storage capacity savings, compression may not be ideally suited for heavily accessed data, or high-performance workloads.
- SmartFlash (L3) caching is not applicable to F810 nodes since they contain exclusively SSD flash media anyway.
- If a heterogeneous cluster contains F810 nodes plus F800 or other non-compression nodes, data will be uncompressed at the time when it moves between pools. A non-compression node on the cluster can be an initiator for compressed writes to an F810 or H5600 pool and will perform compression in software. However, this may generate significant overhead for lower powered Archive class nodes.
- Inline deduplication will not permit block sharing across different hardware types or node pools to reduce the risk of performance asymmetry.
- Inline deduplication will not share blocks across files with different protection policies applied.
- OneFS metadata is not deduplicated.
- Inline deduplication will not deduplicate the data stored in a snapshot.
- There is no inline deduplication of CloudPools files.
- Inline deduplication can deduplicate common blocks within the same file and a sequence of consecutive blocks to a single block in a shadow store, resulting in even better data efficiency.

Efficient storage utilization

Compression is one of several components of OneFS that enable Dell PowerScale to deliver a very high level of storage efficiency. Another enabler is OneFS SmartDedupe, which employs post-process deduplication to share common data blocks across the file system. Other features such as SmartQuotas thin provisioning, SnapshotIQ, and small file packing also contribute to the overall efficiency equation.

However, one of the most significant storage efficiency attributes is the way that OneFS natively manages data protection in the file system. Unlike most file systems that rely on hardware RAID, OneFS protects data at the file level and, using software-based erasure coding, allows most customers to enjoy raw to usable utilization levels of 85% or higher. This is in contrast to the scale up NAS industry mean of around 60% raw disk capacity utilization. Inline data reduction serves to further extend this storage efficiency headroom, bringing an even more compelling and demonstrable TCO advantage to primary file-based storage.

Conclusion

Up until now, traditional compression and deduplication implementations have often been resource intensive, limited to software, and detrimental to performance.

OneFS inline data reduction, integrated with the industry's leading Scale-Out NAS architecture, delivers on the promise of simple data efficiency at scale by providing significant storage cost savings, without sacrificing performance, ease of use or data protection.

With its simple interface and transparent operation, OneFS inline data reduction is easy to manage on your Dell PowerScale cluster, delivering enterprise data efficiency within a single, highly extensible storage pool. Scalability to petabytes and the ability to seamlessly increase capacity and add new technologies, across multiple performance tiers in the same system, means strong investment protection. Integration with OneFS core functions eliminates data risks and gives the user control over what system resources are allocated to data movement.

To learn more about inline data reduction and other storage efficiency products, see Dell PowerScale.

Technical support and resources

Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.

The Dell Technologies Info Hub provides expertise that helps to ensure customer success on Dell storage platforms.